EAI

ELECTRONIC ASSOCIATES, INC.

AFCRL-71-0076

HYBRID COMPUTER APPLICATIONS TO MATHEMATICAL MODELS OF PHYSICAL SYSTEMS – II

Elias H. Hochman
Stuart Mindlin
David Walker

Electronic Associates, Inc.
Scientific Computation Dept.
West Long Branch, New Jersey 07764

Contract No. F 19628-69-C-0286

FINAL REPORT

Period Covered:
June 1, 1969 to October 31, 1969
February 1970

Approved for public release;
distribution unlimited.

Contract Monitor:
Eunice C. Cronin
Analysis and Simulation Branch

Prepared for
AIR FORCE CAMBRIDGE
RESEARCH LABORATORIES
AIR FORCE SYSTEMS COMMAND
UNITED STATES AIR FORCE
BEDFORD, MASSACHUSETTS 01730

72-188

186

Qualified requestors may obtain additional copies from the Defense Documentation Center. All others should apply to the National Technical Information Service.

Unclassified

Security Classification

DOCUMENT CONTROL DATA - R & D

(Security classification of title, body of abstract and indexing annotation must be entered when the overall report is classified)

1. ORIGINATING ACTIVITY (Corporate author) Electronic Associates, Inc. Scientific Computation Department West Long Branch, New Jersey 07764	2a. REPORT SECURITY CLASSIFICATION Unclassified 2b. GROUP

3. REPORT TITLE

HYBRID COMPUTER APPLICATIONS TO MATHEMATICAL MODELS OF PHYSICAL SYSTEMS - II

4. DESCRIPTIVE NOTES (Type of report and inclusive dates)

Scientific. Final. June 1, 1969 to October 31, 1969 Approved 11 Aug. 71

5. AUTHOR(S) (First name, middle initial, last name)

Elias H. Hochman
Stuart Mindlin
David Walker

6. REPORT DATE	7a. TOTAL NO. OF PAGES	7b. NO. OF REFS
February 1970	162	2

8a. CONTRACT OR GRANT NO. F19628-69-C-0286	9a. ORIGINATOR'S REPORT NUMBER(S)
b. PROJECT, TASK, AND WORK UNIT NO. N/A N/A N/A c. DOD ELEMENT 61102F d. DOD SUBELEMENT 681300	9b. OTHER REPORT NO(S) (Any other numbers that may be assigned this report) AFCRL-71-0076

10. DISTRIBUTION STATEMENT

A. Approved for public release; distribution unlimited.

11. SUPPLEMENTARY NOTES	12. SPONSORING MILITARY ACTIVITY
TECH, OTHER	Air Force Cambridge Research Laboratories (SUY) L. G. Hanscom Field Bedford, Massachusetts 01730

13. ABSTRACT

This report describes the applications of hybrid computer techniques to three areas of scientific investigation. The areas investigated were: a variable air density model for a satellite trajectory program, the propagation of electromagnetic energy in the ionosphere, and tracking particles in nuclear emulsions.

The text comprises three separate sections, or reports. Each report includes a discussion of: mathematical modeling, considerations for selecting the hybrid approach, hybrid computer implementation, conclusions and recommendations.

DD FORM 1 NOV 65 1473

Unclassified

Security Classification

Unclassified

Security Classification

14 KEY WORDS	LINK A		LINK B		LINK C	
	ROLE	WT	ROLE	WT	ROLE	WT
Hybrid simulation						
Satellite orbits						
Perturbative factors						
Ray tracing						
Nuclear emulsions						
Track tracing						
Vidicon optics						

Unclassified

Security Classification

AFCRL-71-0076

HYBRID COMPUTER APPLICATIONS TO MATHEMATICAL MODELS OF PHYSICAL SYSTEMS - II

..ias H. Hochman

Stuart Mindlin

David Walker

Electronic Associates, Inc.

Scientific Computation Department

West Long Branch, New Jersey 07764

Contract No. F19628-69-C-0286

FINAL REPORT

Period Covered: June 1, 1969 to October 31, 1969

February 1970

Contract Monitor: Eunice C. Cronin

Analysis and Simulation Branch

Prepared for

Air Force Cambridge
Research Laboratories
AirForce Systems Command
United States Air Force
Bedford, Massachusetts 01730

ABSTRACT

This report describes the application of hybrid computing techniques to three areas of scientific investigation. The areas of investigation were:

I. A Hybrid Computer Air Density Model for a Satellite Trajectory Program

II. Propagation of Electromagnetic Energy in the Ionosphere

III. An Optically Scaled Nuclear Emulsive Track Tracer

The text comprises three separate sections, or reports. Each report includes a discussion of: mathematical modeling, considerations for selecting the hybrid approach, hybrid computer implementation, conclusions and recommendations.

TABLE OF CONTENTS

Preceding page blank

APPENDICES PAGE

I. A HYBRID COMPUTER AIR DENSITY MODEL IMPLEMENTATION FOR A SATELLITE TRAJECTORY PROGRAM

I.1.0 INTRODUCTION

The purpose of this work was to modify the previously existing orbital satellite simulation (Contract No. F19628-67-C-0359) to include a more sophisticated air density model implemented on the hybrid computer. Originally, air density was provided by a digital subroutine which modeled it as being a function only of altitude. This over simplification is unrealistic in that it neglects many other variables that have a signficant effect on the density. Among the more obvious of these are latitude and longitude, and the relative position of the sun.

The new density model that was implemented is taken from U. S. Standard Atmosphere Supplements, 1966, Part 3. Latitude, longitude, hour angle of the sun, day of the year, season, geomagnetic activity, and solar activity effects are included in several formulas which determine the exospheric temperature, T_∞ , i.e. the temperature at the outermost portion of the atmosphere.

T_∞ is used as a term in solving a set of differential equations which yield the number density, n_i, (molecules per cm^3) for each significant atmospheric constituent, as a function of altitude.

The n_i, along with their respective molecular weights m_i, are used to complete the density calculation by a simple algebraic formula.

Implementation of the new model produced a new hybrid SUBROUTINE DENSTY to replace the old digital subroutine of the same name. For each Δt of the digital solution of the satellite motion equations, SUBROUTINE DENSTY is called to compute the atmospheric density.

In the subroutine, T_∞ is first computed on the digital side since the needed formulas are purely algebraic. T_∞ and the altitude, z, are then transferred to the analog which integrates the molecular

density equations. Four such equations are solved in parallel -- one for each of the four most significant gases in the atmosphere: N_2, O, O_2, and He. The molecular densities at altitude z are transferred back to the digital which completes the density calculations.

In addition a density check subroutine was added to the program which, when called, performs a dynamic check of the entire hybrid density model to confirm that it is operating both correctly and accurately.

The new hybrid orbiting satellite simulation program is called SIMSAT II.

I.2.0 CALCULATION OF EXOSPHERIC TEMPERATURE

In order to solve for the number densities leading to the air density, it is necessary to compute the exopsheric temperature, T_∞ ; i.e. the temperature at the outermost portion of the atmosphere. It is T_∞ which is modeled as a function of all the parameters, except altitude, which influence density. The effect of each parameter is treated separately.

I.2.1 VARIATION WITH SOLAR ACTIVITY

The parameter used to describe variation of T_∞ due to solar activity is the 10.7-centimeter solar flux which is monitored by the National Research Council in Ottawa, Ontario. Effects of the slow 11-year cycle variation and the 27-day solar rotation are both considered.

I.2.1.1 VARIATION WITH THE SOLAR CYCLE

Let $\bar{F}_{10.7}$ be the 10.7-centimeter solar flux in units of 10^{-22} watts/m^2/cycle/sec and $\bar{T}_0$ the night time global minimum value

of T_∞ , each averaged over three solar rotations. The formula relating these two quantities for quiet geomagnetic conditions is:

$$\bar{T}_o = 362 + 3.60\ F_{10.7}$$

I.2.1.2 VARIATION WITHIN ONE SOLAR ROTATION

Let $F_{10.7}$ be the daily mean of the 10.7-centimeter solar flux. We can correct $\bar{T}_o$ for the day-to-day temperature variation due to the variation within one solar rotation. The formula:

$$T_o' = \bar{T}_o + 1.8\ (F_{10.7} - \bar{F}_{10.7})$$

yields T_o' , the corrected night time global minimum of T_∞. Values of $F_{10.7}$ and $\bar{F}_{10.7}$ at orbit insertion are kept constant for the lifetime of the satellite.

I.2.2 SEMI-ANNUAL VARIATION

The semiannual variation is described by the function

$$f(d) = \left(0.37 + 0.14 \sin 2\pi \frac{d-151}{365}\right) \sin 4\pi \frac{d-59}{365}$$

where d is the number of days elapsed since January 1 of each year.

This variation is used to produce T_o, the night time global minimum exospheric temperature according to the formula

$$T_o = T_o' + f(d)\ F_{10.7}$$

The combination of sine terms produce two unequal pairs of maxima and minima. The primary and secondary maxima occur on October 14 and April 20 respectively; and the primary and secondary minima occur on July 18 and January 8 respectively.

Note that the solar and semiannual variations can be combined by substitution into the single formula:

$$T_o = 362 + 1.8\ F_{10.7} + (1.8 + f(d))\ F_{10.7}$$

I.2.3 DIURNAL VARIATION

The distribution of exospheric temperature on the earth is such that the maximum occurs at 1400 hours local solar time. The atmosphere bulges out in the bright hemisphere (the durnal bulge), producing higher densities above 200 km. The center of the bulge is at latitude ϕ_B.

The maximum global exospheric temperature, T_x, and minimum global exospheric temperature, T_o, are related by the formula

$$\frac{T_x}{T_o} = 1 + R$$

where R has been found to be 0.28.

For latitude ϕ two new angles as defined:

$$\eta = \tfrac{1}{2}\left|\phi - \phi_B\right| \quad \text{and} \quad \theta = \tfrac{1}{2}\left|\phi + \phi_B\right|$$

ϕ_B is taken to be zero; i.e. the diurnal bulge is modeled as residing on the equation. Thus $\eta = \theta = \tfrac{1}{2}\left|\phi\right|$.

To represent the effect of the sun's position, an angle τ must be computed:

$$\tau = H^* + \beta + p \sin (H^* + \nu)$$

The constants used in the above formula are assigned values as follows:

$$\beta = -45^{\circ}$$
$$p = 12^{\circ}$$
$$\gamma = +45^{\circ}$$

H^{*} is the hour angle of the sun measured from its most overhead position, i.e. 1200 hours local solar time. The relationship between hour angle and local solar time is illustrated by Figure (I-1)

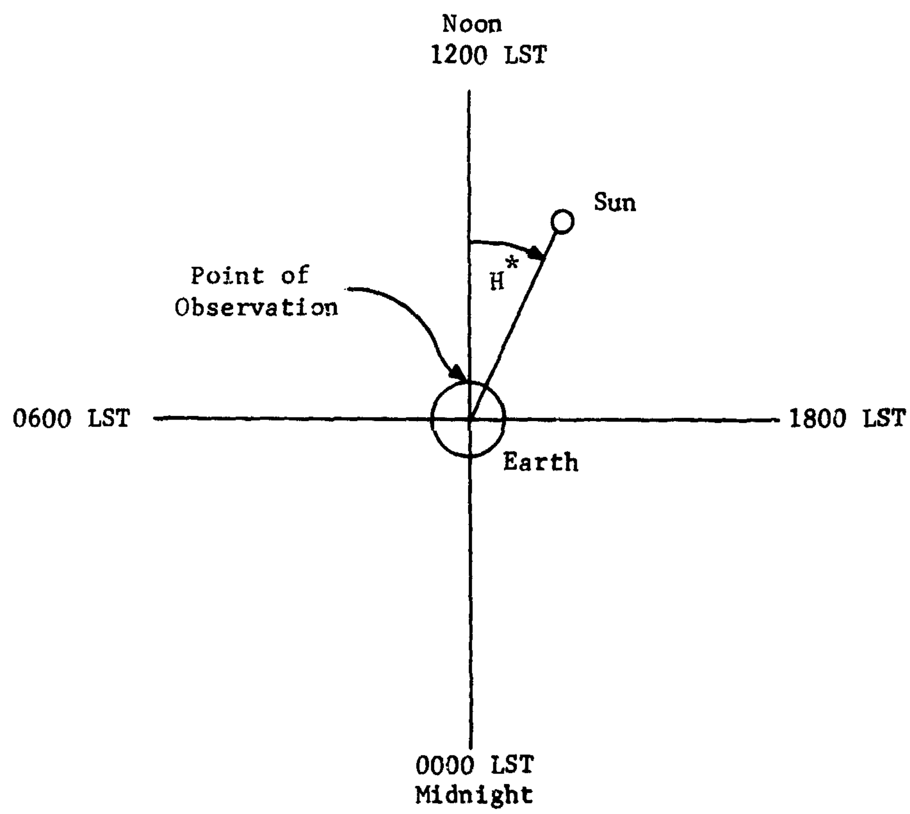


FIGURE (I-1)

The diurnal variation can now be computed from the formula:

$$\frac{T}{T_o} = \left(1 + R \sin^m \theta\right)\left(1 + A \cos^n \frac{\tau}{2}\right)$$

where $A = R \dfrac{\cos^m \eta - \sin^m \theta}{1 + R \sin^m \theta}$

$m = 1.5$

$n = 2.5$

I.2.4 VARIATION WITH GEOMAGNETIC ACTIVITY

In order to complete the calculation of exospheric temperature, a ΔT due to geomagnetic influence must be added to the already computed T.

This is found from the three-hourly geomagnetic planetary index, K_p, by the formula:

$$\Delta T = 28 K_p + 0.03 e^{K_p}$$

Note that $\Delta T \neq 0$ for $K_p = 0$ and increases with K_p. Since temperature variation lags behind K_p by about 7 hours, the value of K_p 7 hours before orbit insertion time is used in the model and is kept constant for the lifetime of the satellite.

The computation of exospheric temperature is completed by the addition

$$T_\infty = T + \Delta T$$

I.3.0 CALCULATION OF AIR DENSITY

Having computed the exospheric temperature, the density for any altitude can be found by integrating the gas equations. The solutions to these equations produce air densities that are considered valid

for the spring and fall season. For the summer and winter seasons, modifications must be made to complete the density model.

I.3.1 THE DIFFUSION EQUATIONS

Diffusive equilibrium is assumed above 120 K_m. Using the ideal gas law with a diffusion term, the diffusion equation is for the number density, n_i, is written as:

$$\frac{dn_i}{n_i} = - \frac{m_i g}{K} \frac{dz}{T} - \frac{dT}{T} (1 + \alpha) \qquad (1)$$

Here, α is the thermal-diffusion factor, K is the Boltzmann constant, and g is the acceleration of gravity. m_i is the molecular weight of the constituent.

The temperature follows an exponential rise from 120 km to infinity:

$$T = T_\infty - (T_\infty - T_{120}) e^{-s(z-120)}$$

This is the solution of the differential equation:

$$\frac{dT}{dz} = -s(T - T_\infty) \qquad (2)$$

Multiplying equation (1) by $\frac{n_i}{dz}$ and substituting equation (2) yields:

$$\frac{dn_i}{dz} = - \frac{n_i m_i g}{KT} + \frac{n_i \; (1 + \alpha)s}{T} (T - T_\infty) \qquad (3)$$

The value of α for helium has been found, by Kockarts and Nicolet, to be -0.38. For N_2, O, and O_2 it was assumed that $\alpha = 0$.

For all values of T_∞ the boundary values are fixed as follows:

$$T_{120} = 355.0^{\circ}K$$
$$n(N_2) = 4.0 \times 10^{11}\ cm^{-3}$$
$$n(O_2) = 7.5 \times 10^{10}\ cm^{-3}$$
$$n(O) = 7.6 \times 10^{10}\ cm^{-3}$$
$$n(He) = 3.4 \times 10^{7}\ cm^{-3}$$

s is a function of T_∞ and is found by the formula:

$$s = 0.0291 \exp(-q^2/2) \text{ in units 1/km}$$

$$\text{where } q = \frac{T_\infty - 800}{750 + 1.722 \times 10^{-4}\ (T_\infty - 800)^2}$$

Because great variations in altitude are involved with these calculations, the acceleration of gravity, g, is not taken as being constant but is found from the inverse square formula:

$$g = \frac{g_o R_o^{\ 2}}{(R_o + z)^2}$$

Where g_o is the value of g at the earth's surface and R_o is the earth's radius, taken to be 6371 km.

Integrating the set of four equations (3) from 120 km to altitude z yields the needed n_i.

The density, ρ, is then found by the simple formula

$$\rho = \frac{1}{A}\sum_{i=1}^{4} n_i m_i$$

where A is Avogadro's number, 6.023×10^{23} molecules/mole.

I.3.2 THE SUMMER AND WINTER MODIFICATIONS

It is found that the densities computed from the diffusion equations are accurate representations of the behavior during spring and fall.

However, modifications must be made for the other two seasons because the air at the altitudes considered is less dense in summer and more dense in winter.

The deviations from the spring-fall models are illustrated for three values of T_∞ by the chart in Figure (I-2) taken from Page 40 of U. S. Standard Atmosphere Supplements, 1966.

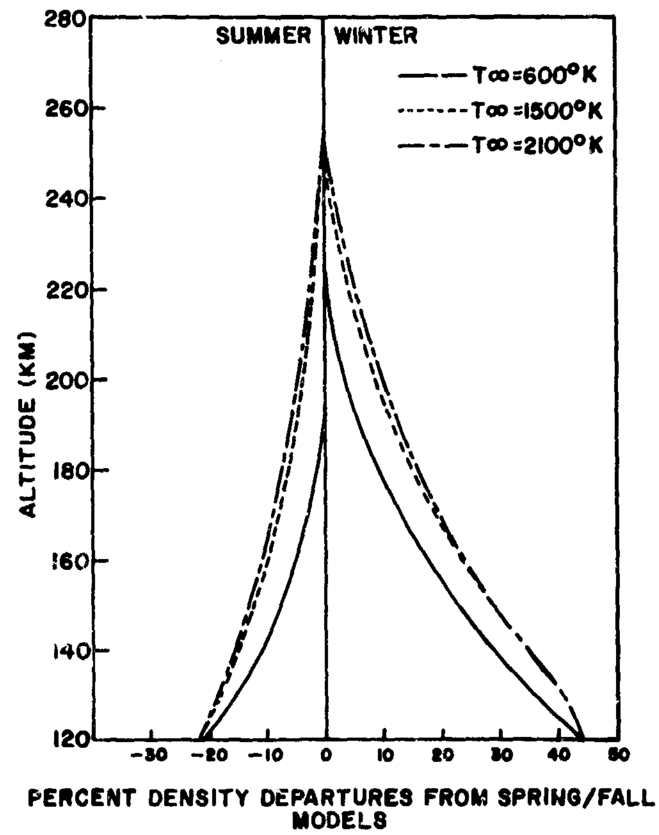


FIGURE (I-2)-Density departures from the spring/fall models for summer and winter with three exospheric temperatures.

Note that at higher altitudes, the winter and summer densities merge into the spring-fall density. The lowest altitude that the models first match, Z_m, is a function of T_∞ and are taken from Table 1 in the reference and are listed in Table (I-1).

TABLE (I-1)

T_∞ °K	Z_m (Summer) km	Z_m (Winter) km
600	195	220
700	200	225
800	210	230
900	220	235
1000	230	240
1100	235	240
1300	245	245
1500	250	250
1700	255	255
1900	255	255
2100	255	255

Below Z_m, the winter density, ρ_w, and the summer density, ρ_s, are found from the spring-fall density, ρ_j by multiplying by the factors found in the formulas below:

$$\rho_w = \rho_j W_r$$

$$\rho_s = \rho_j S_r$$

where $W_r = (1.4848 - .4848\psi)$

$S_r = (.7919 + .2081\psi)$

and $\psi = \tanh 1.75D + .059D^2$

$$D = \frac{z-120}{Z_m-120}$$

I.4.0 ORGANIZATION OF THE HYBRID PROGRAM

The air density model described was incorporated into the orbiting satellite program developed previously by EAI for AFCRL under Contract Number F19626-67-C-0359. The all-digital SUBROUTINE DENSTY of that program, which modeled density only as a function of altitude, was replaced by a new hybrid SUBROUTINE DENSTY to implement the more realistic model described in this report.

The day of the year and Greenwich Mean Time of orbit insertion are read in on cards, as are both the daily and monthly means of the 10.7-centimeter solar flux. Since SUBROUTINE INITAL read all cards in the original program, these new read statements were added to that subroutine. SUBROUTINE INITAL also calcuates ΔT from the value of K_p read in on a card. Initialization of all variables used in the density calculation is done by this subroutine.

I.4.1 SUBROUTINE DENSTY

For each time step of digital integration SUBROUTINE DENSTY is called by the main program. The subroutine first checks if satellite altitude z is less than BRNOUT, the minimum altitude before satellite burn-out. If it is, a variable IBNOUT is set to 1, and the main program halts the simulation. If the satellite has not burned out, the subroutine proceeds with the density calculation.

First the day of the year is calculated from the initial day and the total elapsed time. The exopheric temperature T_∞ is then calculated using all the algebraic formulas described in Section 2. This is used to compute the value of s. Then T_∞, s, and the satellite altitude z are sent to the analog computer through digital-to-analog converters.

A control line is reset which puts the analog circuits in the operate mode. The diffusion equations are then integrated from 120 km to z'. (The prime is used to distinguish satellite altitude from the independent variable, z, on the analog computer.) The values of n_i for N_2, O_2, O, and He are transferred back to the digital computer by analog-to-digital converters and the analog circuits are reset.

The spring-fall density is then calculated as a weighted sum using the molecular weights and Avogadro's number.

The proper summer or winter modification is multiplied if the season is not spring or fall. A variable, ICT, indicates the number of days elapsed in the present season of the year. After ICT reaches 91 the subroutine changes the season, which is represented by the variable ISN; the seasons summer, fall, winter and spring being represented by the values 1-4 respectively. Note that a seasonal change (for example from fall to winter) would add an "instantaneous" jump in density if the modification factor were applied directly. In order to smooth out this jump in density the winter and summer modifications are weighted by a smooth curve that applies them such that there is no effect at the beginning of the season, increases to the maximum at the middle of the season, and decreases to no modification again at the end of the season. This "modification to the modification" is described by a second order algebraic equation (parabola) and is illustrated in Figure (I-3).

A detailed flow chart and list of variable definitions for SUBROUTINE DENSTY are given in Appendix IA.5.

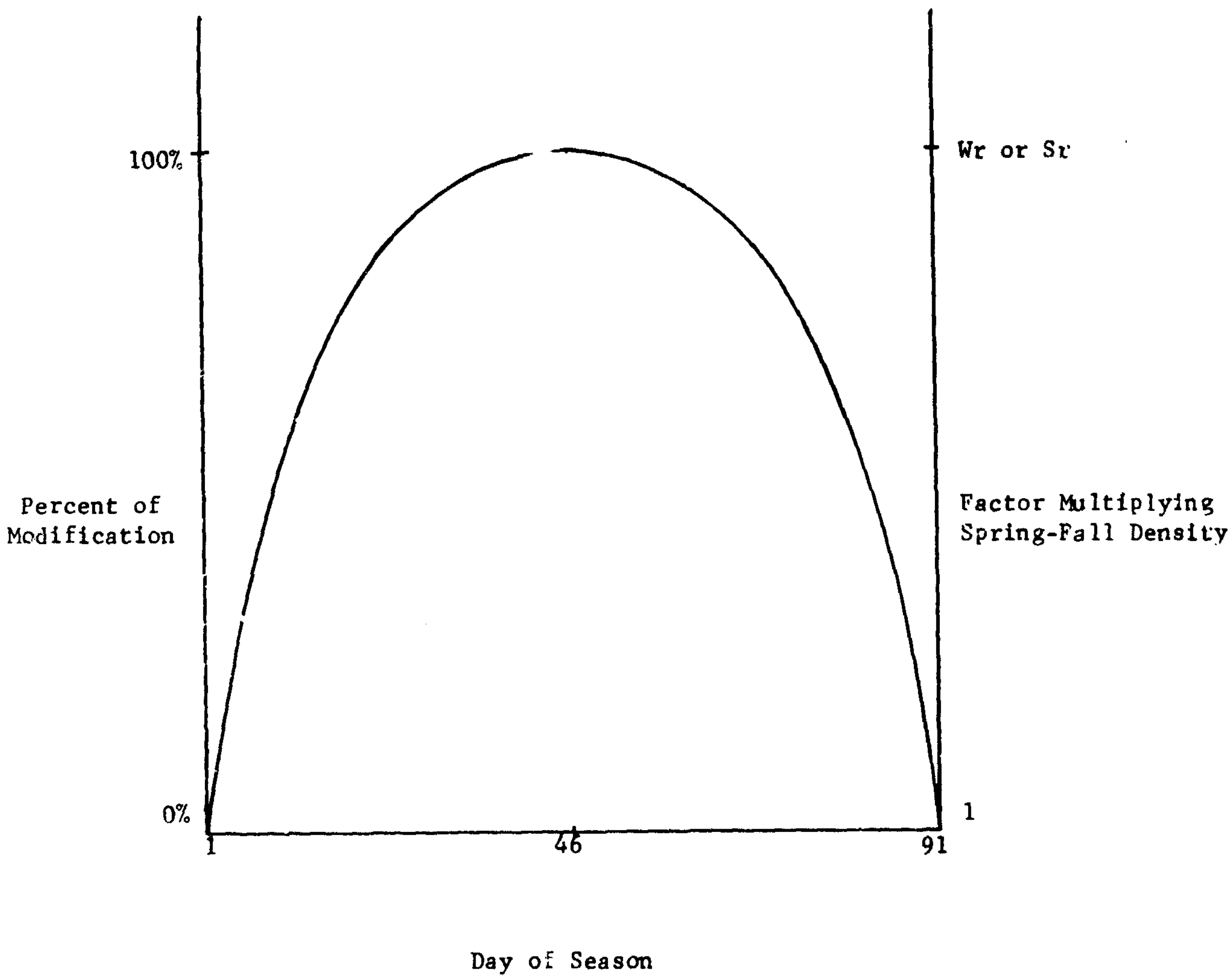


Figure (I-3) Winter and Summer Modification Smoothing Function

I.5.0 MODEL VERIFICATION

An automatic pot setting subroutine and static check subroutine have been incorporated into the new hybrid orbiting satellite program.

In addition, a complete dynamic check of the density model is available to the user.

At the beginning of each run, the operator can initiate the dynamic density check by setting a console switch. The analog integrates the diffusion equations up to a number of different altitudes for each of several values of T_∞. The log n_i are returned to the digital and the density computed for each case.

The results are displayed on the line printer in the same format as the tables at the end of the U. S. Standard Atmosphere Supplements book. Note that the units have been changed for the printout to agree with the entries in the tables. This allows for ready comparison of values to verify that the model is operating properly, and if not, help determine the source of the problem.

I.6.0 CONCLUSIONS AND RECOMMENDATIONS

The hybrid air density model implemented is a great improvement over pure altitude functions for use in low altitude satellite simulations. By integrating the diffusion equations at high speed on the analog computer, the digital computer can still integrate the satellite trajectory equations accurately and at speeds as high as 1000 time real time.

In addition, the time step is automatically adjusted to be smaller when the satellite is at lower altitudes. Thus maximum accuracy is achieved at the portion of the orbit where air density is most significant.

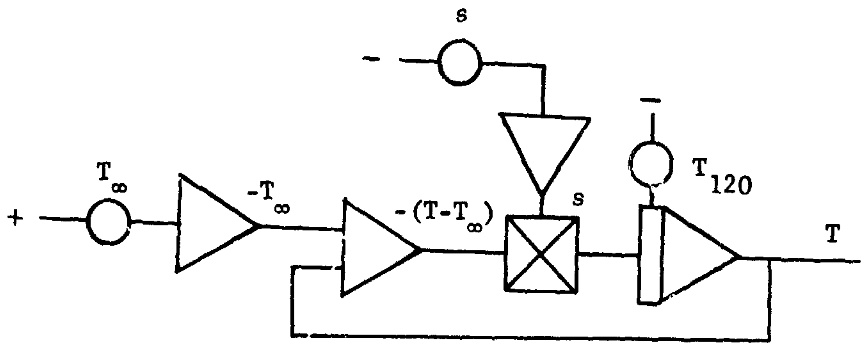
The additional parameters included insure a more realistic representation of the atmosphere and its influence on satellite motion. However, to further improve the density model it is recommended that a variable representation be included for the parameters $F_{10.7}$, $F_{10.7}$, and K_p. In the present version of the program, the values of these three parameters at orbit insertion are kept constant for the lifetime of the satellite. Since these parameters are continuously varying an improvement would be realized by replacing them with time dependent functions which would better represent their behavior.

APPENDIX I

IA1 AMPLITUDE SCALING

IA1.1 TEMPERATURE EQUATION

The equation for temperature variation along with its unscaled analog circuit diagram are shown below:



Equation: $\frac{dT}{dz} = -s(T-T_\infty)$

The scaling table is:

Problem Variable	Estimate Maximum -	Computer Variable
T	2500^oK	$\left[\frac{T}{2500}\right]$
s	1/30 km^{-1}	$\left[30s\right]$

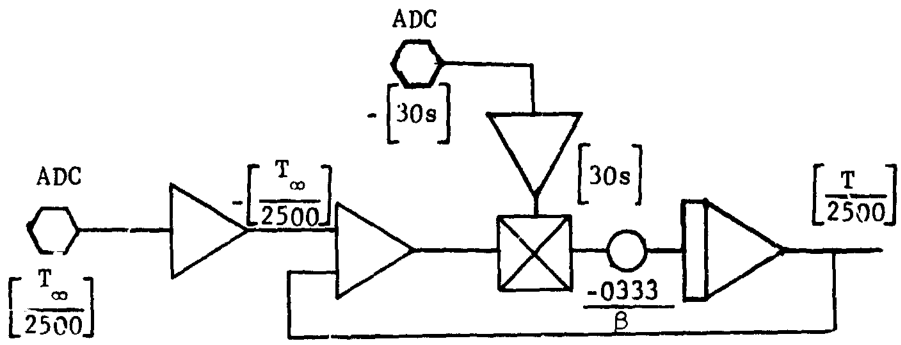
The scaled equation is now formed:

$$- \frac{dT}{dz} = s\ (T - T_\infty)$$

$$-2500 \frac{d}{dz}\left[\frac{T}{2500}\right] = \frac{2500}{30}\ \left[30s\right] \left\{\left[\frac{T}{2500}\right] - \left[\frac{T_\infty}{2500}\right]\right\}$$

$$- \frac{d}{dz}\ \left[\frac{T}{2500}\right] = .0333 \left[30s\right] \left\{\left[\frac{T}{2500}\right] - \left[\frac{T_\infty}{2500}\right]\right\}$$

From this equation the scaled diagram is drawn



A1.2 INVERSE SQUARE GRAVITATIONAL VARIATION

The variation in gravitational acceleration with altitude is given by:

$$g = g_o\ V_{is}$$

$$V_{is} = \frac{Ro^2}{(Ro + z)^2} \quad \text{with } R_o = 6371 \text{ cm}$$

The scaling table is

Problem Variable	Estimated Maximum	Computer Variable
z	1000 km	$\left[\frac{z}{1000}\right]$
V_{is}	1	$\left[V_{is}\right]$

The scaled equation is then developed.

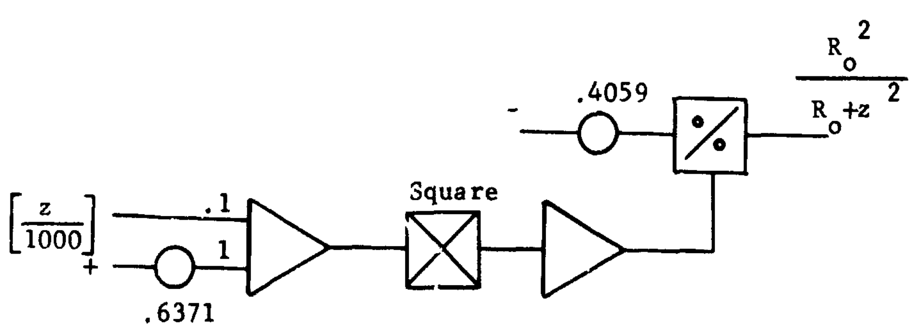
$$V_{is} = \frac{Ro^2}{\left(Ro + 1000\left[\frac{z}{1000}\right]\right)^2}$$

$$V_{is} = \frac{Ro^2}{10^8\left(\frac{Ro}{10{,}000} + .1\left[\frac{z}{1000}\right]\right)^2}$$

Substituting 6371 for Ro yields

$$V_{is} = \frac{.4059}{\left(.6371 + .1\left[\frac{z}{1000}\right]\right)^2}$$

The scaled diagram is thus:



IA1.3 DIFFUSION EQUATIONS

The set of diffusion equations to be solved are repeated here for convenience:

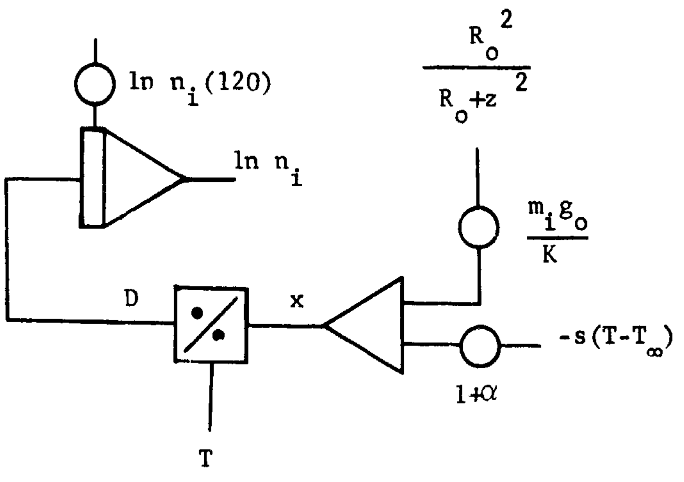
$$\frac{dn_i}{dz} = - \frac{n_i m_i g}{KT} + \frac{n_i}{T} (1 + \alpha) s (T - T_\infty)$$

Because of scaling considerations, it is preferable to work with $\ln n_i$. From the basic relation $\frac{d}{dn_i} \ln n_i = \frac{1}{n_i}$ it follows that $dn_i = n_i \ln n_i$. Substituting in the above diffusion equations yields, after canceling n_i:

$$\frac{d}{dz} (\ln n_i) = - \frac{m_i g}{KT} + \frac{(1 + \alpha) s}{T} (T - T_\infty)$$

It is this final set of equations which is solved on the analog computer.

The unscaled diagram that describes the four diffusion equations is shown below:



Note that for scaling purposes we have defined the outputs of the summer and divider as x and D respectively

In order to be consistant with the table entries in U. S. Standard Atmosphere Supplements, 1966, the common logarithm is substituted for the natural logarithm according to the formula:

$$\ln n = 2.3026 \log n$$

The unscaled diffusion equation now becomes:

$$2.3026 \frac{d}{dz} (\log n) = - \frac{m_i g}{KT} + \frac{(1 + \alpha)\ s\ (T - T_\infty)}{T}$$

The scaling table is:

Problem Variable	Estimated Maximum	Computer Variable
T	$2500^\circ K$	$\left[\frac{T}{2500}\right]$
s	$\frac{1}{30}$ km^{-1}	$\left[30s\right]$
x	$100^\circ K/Km$	$\left[\frac{x}{100}\right]$
D	$\frac{1}{3}$ km^{-1}	$\left[3D\right]$
log n	30	$\left[\frac{\log n}{30}\right]$

The scaled equations are now formed:

$$x = - \frac{m_i g}{K} - (1 + \alpha)\ s\ (T - T_\infty)$$

$$100 \left[\frac{x}{100}\right] = - \left(\frac{m_i g}{K} - \frac{2500}{30} (1 + \alpha) \left[30s\right] \left\{ \left[\frac{T}{2500}\right] - \left[\frac{T_\infty}{2500}\right] \right\} \right)$$

$$\frac{x}{100} = - \left(\frac{m_i g}{100K} - .8333\ (1 + \alpha) \left[30s\right] \left\{ \left[\frac{T}{2500}\right] - \left[\frac{T_\infty}{2500}\right] \right\} \right)$$

$$D = \frac{x}{T}$$

$$\frac{1}{3}\left[3D\right] = \frac{100\left[\frac{x}{100}\right]}{2500\left[\frac{T}{2500}\right]}$$

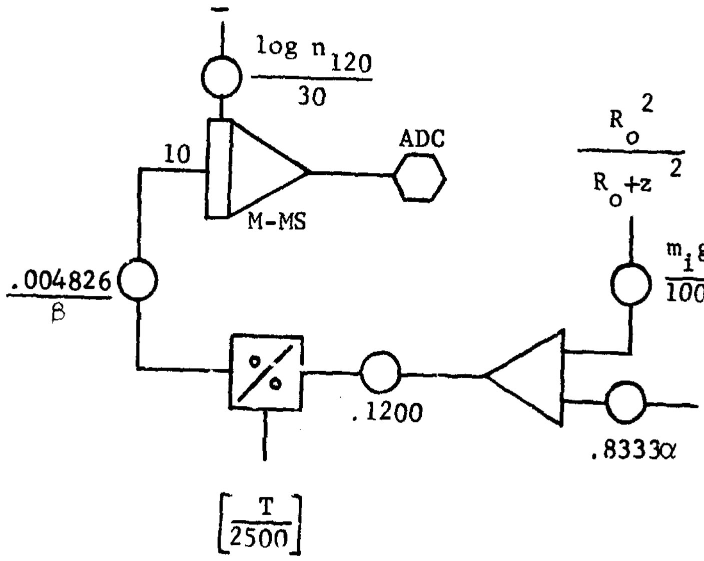
$$3D = \frac{.12\left[\frac{x}{100}\right]}{\left[\frac{T}{2500}\right]}$$

$$-2.3026\,\frac{d}{dz}\,(\log n) = D$$

$$-30(2.3026)\,\frac{d}{dz}\left[\frac{\log n}{30}\right] = \frac{1}{3}\left[3D\right]$$

$$\frac{-d}{dz}\left[\frac{\log n}{30}\right] = .004826\left[3D\right]$$

From these equations the scaled diagram is drawn:



IA1.4 ALTITUDE RAMP

A circuit is necessary to integrate from 120 km up to the satellite altitude z'.

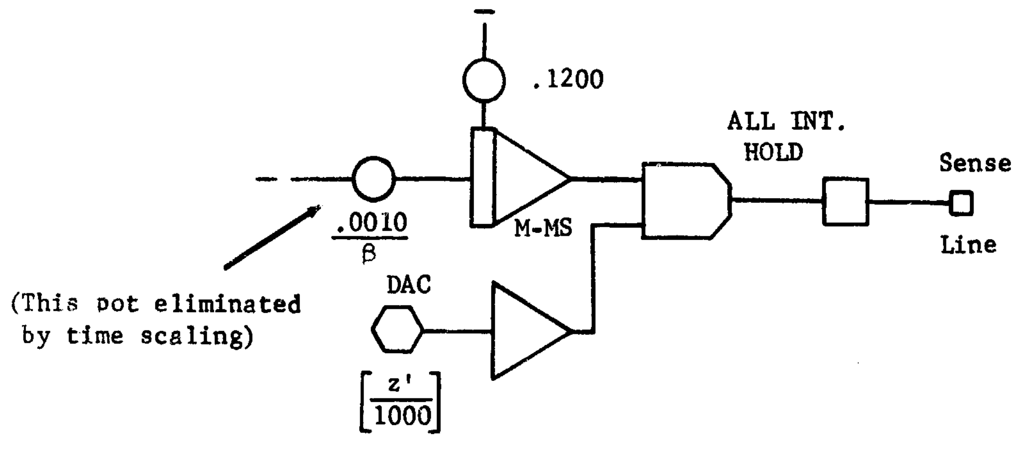
The unscaled equation is merely:

$$\frac{dz}{dz} = 1$$

But z has a scale factor of 1000 so this becomes

$$\frac{d}{dz}\left[\frac{z}{1000}\right] = .001$$

The scaled diagram is:



IA2. "TIME" SCALING OF THE ANALOG COMPUTER

During each Δt of digital integration, the digital computer waits for the analog computer to integrate the diffusion equations to obtain density. Therefore to keep Δt small to insure high accuracy it is advantageous to run the analog computer as fast as possible.

A value for β of 10^{-5} is chosen because it provides the best combination of speed and accuracy.

According to the formula

$$t = \beta z$$

This means that z will integrate 1000 km in 1 millisecond.

IA3. ANALOG COMPUTER CIRCUIT DIAGRAMS

The complete scaled diagrams for the hybrid density model are shown in figures IA1 and IA2. Values of potentiometer settings and static check values for these circuits are given in Tables IA1 and IA2 respectively.

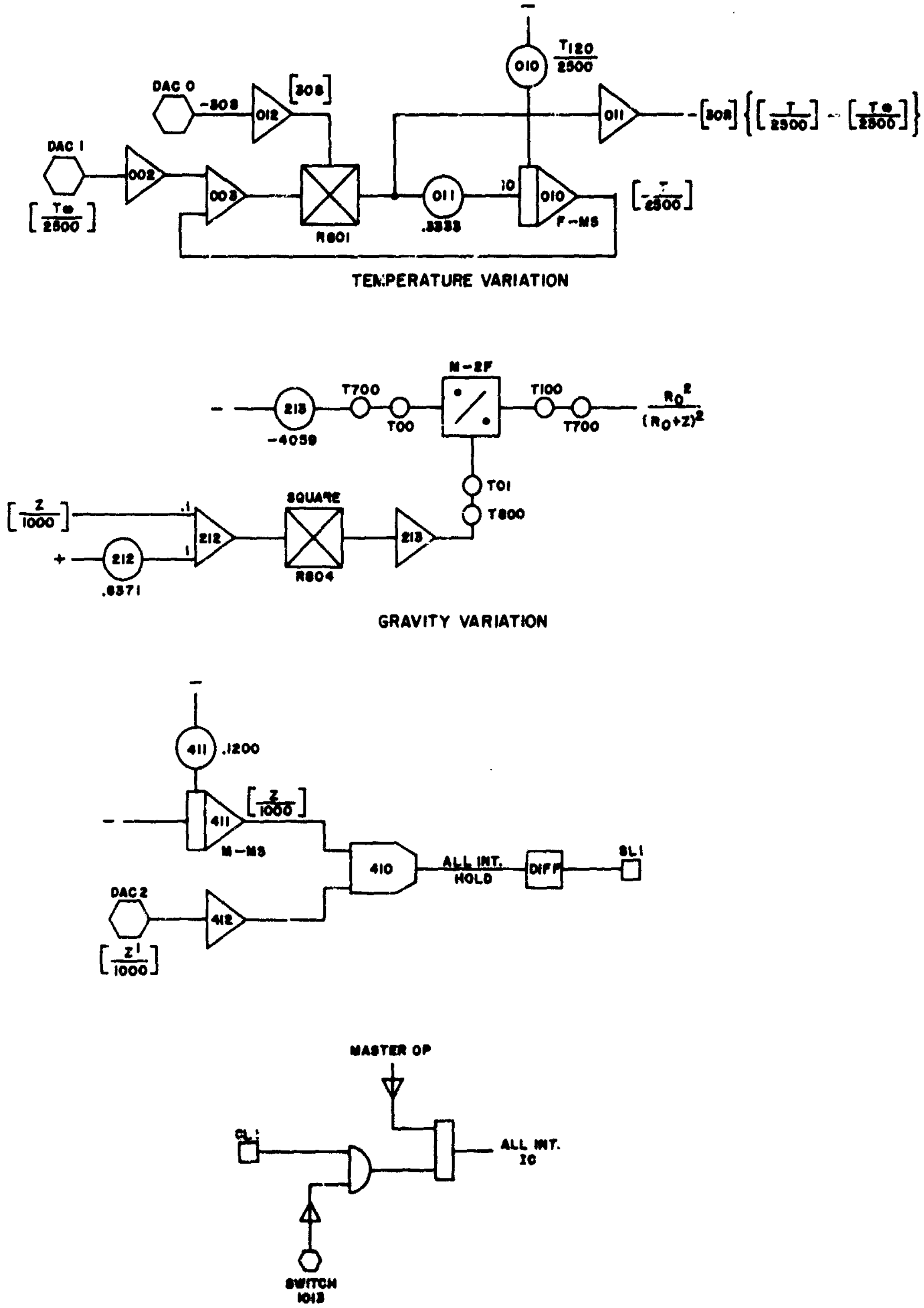


Figure IA1 Analog Circuits

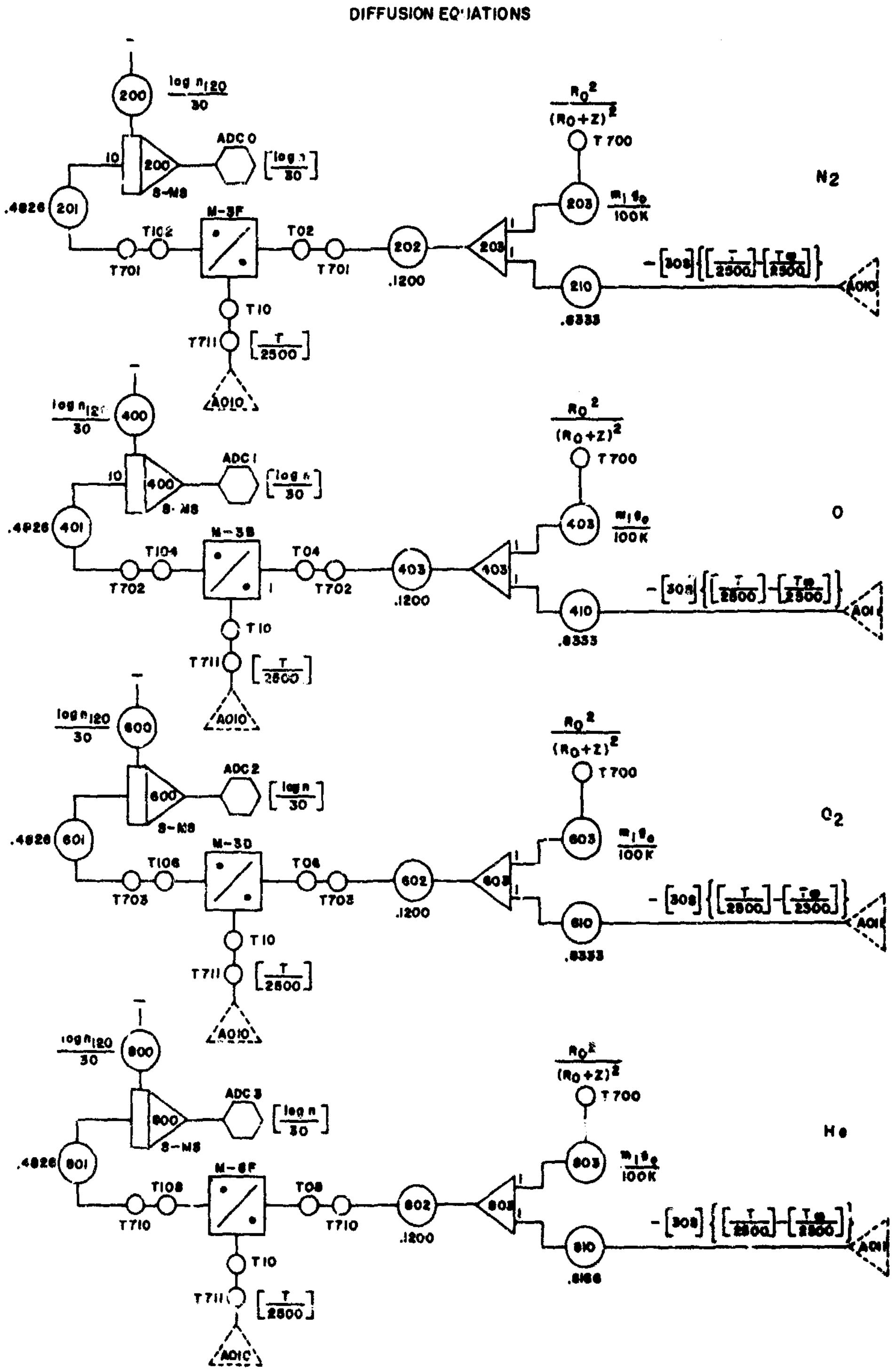


Figure IA2 Analog Circuits

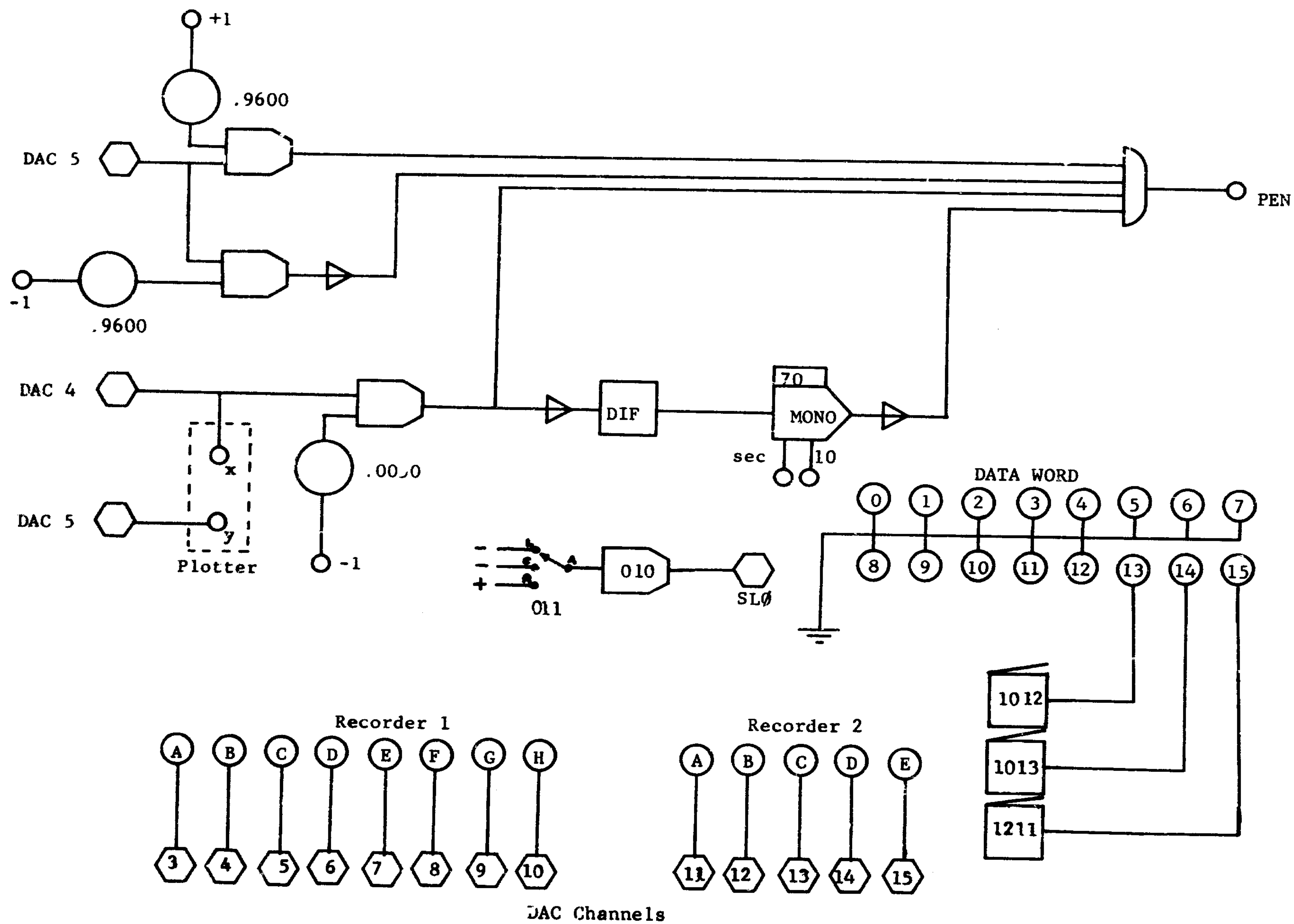


FIGURE IA3 LOGIC DIAGRAM

Calculation of $\frac{m_i g_o}{K}$

$m_i = m_w$ grams/mole

where m_w is molecular weight as a pure number

$g_o = .0098$ km/sec^2

$k = 1.3805 \times 10^{-26}$ gm km^2/sec^2 ^{o}K molecule

Using above

$$\frac{m_i\ g_o}{K} = 7.098 \times 10^{23}\ m_w\ \frac{\text{molecules}\ ^oK}{\text{mole km}}$$

We can simplify the units by dividing by Avogadro's number 6.02 x 10^{23} molecules/mole which yields

$$\frac{m_i\ g_o}{K} = 1.18\ m_w\ \frac{^oK}{km}$$

Using this formula, the following table of values is formed

Atmospheric Contituent	M_w	$\frac{m_i g}{K}$ (^{o}K/km)
N_2	28	33.05
O	16	18.88
O_2	32	37.76
He	4	4.72

Pot Number		Parameter	Setting
C010		$\frac{T_{120}}{2500}$	.1420
C011		$\frac{.00333}{\beta}\ 10^{-4}$	.3333
C212		$\frac{R_o}{10^4}$	.6371
C213		$\frac{R_o^{\ 2}}{10^8}$	.4059
C411		$\frac{120}{1000}$	.1200
C200		$\frac{\log n_{120}}{30}$	.8867
C201		$\frac{.0004826}{\beta}\ 10^{-3}$	.4826
C202	N_2	(scaling)	.1200
C203		$\frac{m_i g_o}{100K}$	.3305
C210		$.8333\ (1+\alpha)$	.8333
C400		$\frac{\log n_{120}}{30}$	.8627
C401		$\frac{.0004826}{\beta}\ 10^{-3}$	.4826
C402	O	(scaling)	.1200
C403		$\frac{m_i g_o}{100K}$	.1888
C410		$.8333\ (1+\alpha)$	.8333

TABLE (IA1)

POTENTIOMETER SETTINGS
FOR DENSITY MODEL

Pot Number		Parameter	Setting
C600		$\frac{\log n_{120}}{30}$	.8625
C601		$\frac{.0004826}{\beta} \quad 10^{-3}$	.4826
C602	O_2	(scaling)	.1200
C603		$\frac{m_i \, g_o}{100K}$	.3776
C610		$.8333 \, (1 + \alpha)$	.8333
C800		$\frac{\log n_{120}}{30}$	.7510
C801		$\frac{.0004826}{\beta} \quad 10^{-3}$	.4826
C802	He	(scaling)	.1200
C803		$\frac{m_i \, g_o}{100K}$	.0472
C810		$.8333 \, (1 + \alpha)$	.5166

TABLE (IA1) CONT'D

STATIC CHECK

Component	Value
A002	-.4000
A003	.2580
A012	.8430
R801	-.2175
P011	-.0725
P010	-.1420
A010	.1420
A011	.2175
P212	.6371
A212	-.6491
R804	-.4213
A213	.4213
P213	-.4059
M-2F(231-R)	.9634
P411	-.1200
A411	.1200
P200	-.8867
A200	.8867
P201	.2039
M-3F(231-R)	.4225
P202	-.0600
A203	-.4996
P203	.3184
P210	.1812
P400	-.8627
A400	.8627
P401	.1482
M-3B(231-R)	.3070
P402	-.0436
A403	-.3631

TABLE (IA2)
STATIC CHECK VALUES
FOR DENSITY MODEL

Component	Value
P403	.1819
P410	.1812
P600	-.8625
A600	.8625
P601	.2223
M-3D (231-R)	.4606
P602	-.0654
A603	-.5450
P603	.3638
P610	.1812
P800	-.7510
A800	.7510
P801	.0642
M-8F (231-R)	.1331
P802	-.0189
A803	-.1579
P803	.0455
P810	.1124

TABLE (IA2) CONT'D

IA4. VARIABLE DEFINITIONS FOR SUBROUTINE DENSTY

Z	-	satellite altitude	
R	-	orbital radius	
RØ	-	radius of earth	
BRNOUT	-	minimum satellite altitude before burn-out	
IBNOUT	-	set to 1 if satellite has burned out	
D	-	present day of the year	
IPD	-	integar part of D	
DØ	-	day of year of orbit insertion	
T	-	elapsed time in seconds from orbit insertion	
ICT	-	number of days elapsed in present season	
F	-	semiannual variation function	
TØ	-	night time global minimum exospheric temperature	
GMT	-	present Greenwich Mean Time	
GMTØ	-	Greenwich Mean Time at orbit insertion	
LAMN	-	longitude of satellite	
LST	-	local solar time	
LN	-	latitude of satellite	
ETA	-	η	
THETA	-	θ	diurnal bulge angles
TAU	-	π	
H	-	hour angle of the sun (H^*)	
A	-	intermediate values in diurnal	
RATIO	-	variation computation	
TNF	-	exospheric temperature (T_∞)	
TNFDLT	-	geomagnetic variation (ΔT)	
TM	-	(T_∞ - 800)	
Q	-	temperature constants	
S	-		
DA(3)	-	digital to analog conversion array	
LOGN(4)	-	analog to digital conversion array	
RHO	-	air density (ρ)	

ISN	-	season of the year indicator
RNTNF	-	winter-summer matching altitude pointer
NTNF	-	integer part of RNTNF
ZMS(NTNF)	-	summer altitude matching table
ZMW(NTNF)	-	winter altitude matching table
ZM	-	matching altitude, Z_m
E	-	intermediate value in summer-winter modification
PSI	-	ψ, summer-winter modification parameter
SR	-	summer modification (Sr)
WR	-	winter modification (Wr)
A,B,C	-	constants for parabolic smoothing function
SRMOD	-	modification to summer modification
WRMOD	-	modification to winter modification

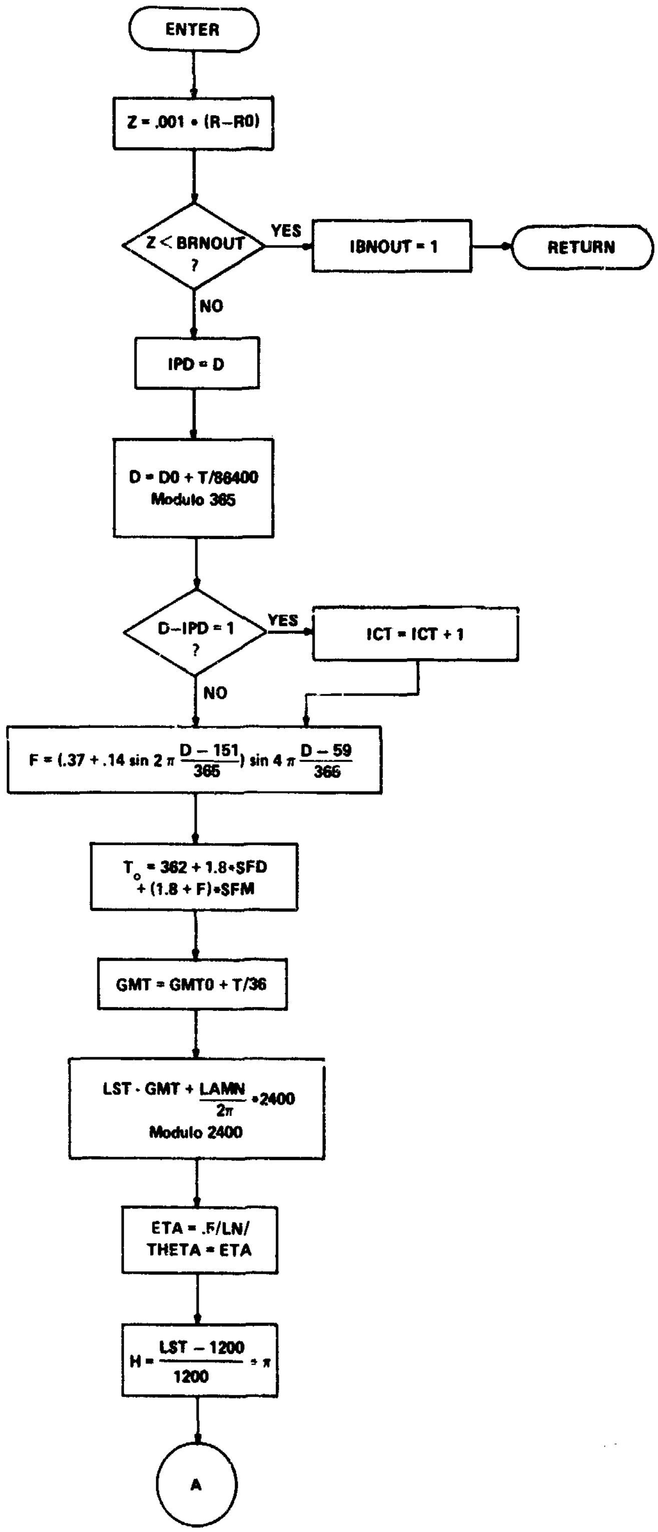


Figure (IA5.) Flowchart for Subroutine Density

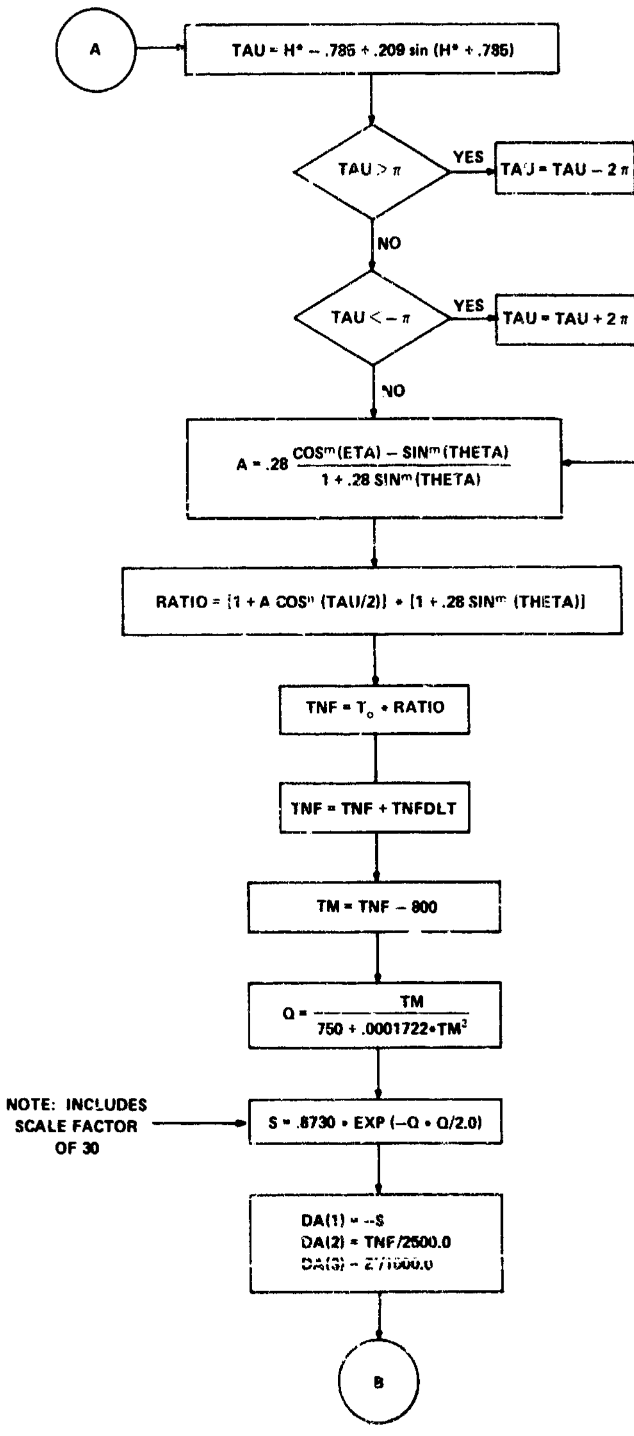


Figure (IA5.) Flowchart for Subroutine Density, Cont.

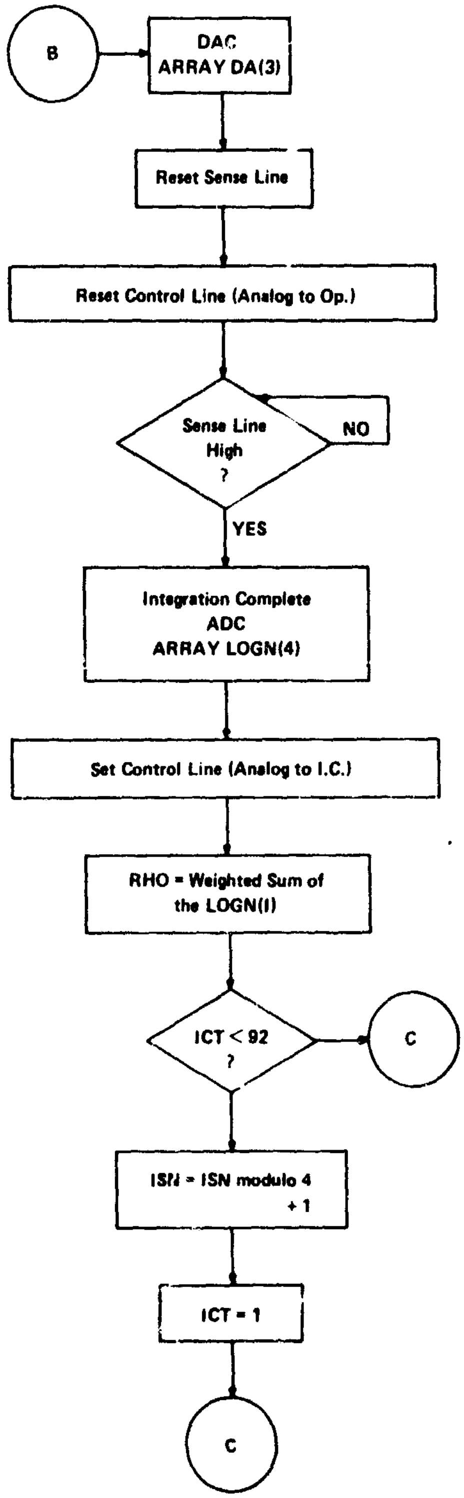


Figure IA5 Flowchart for Subroutine Density, Cont.

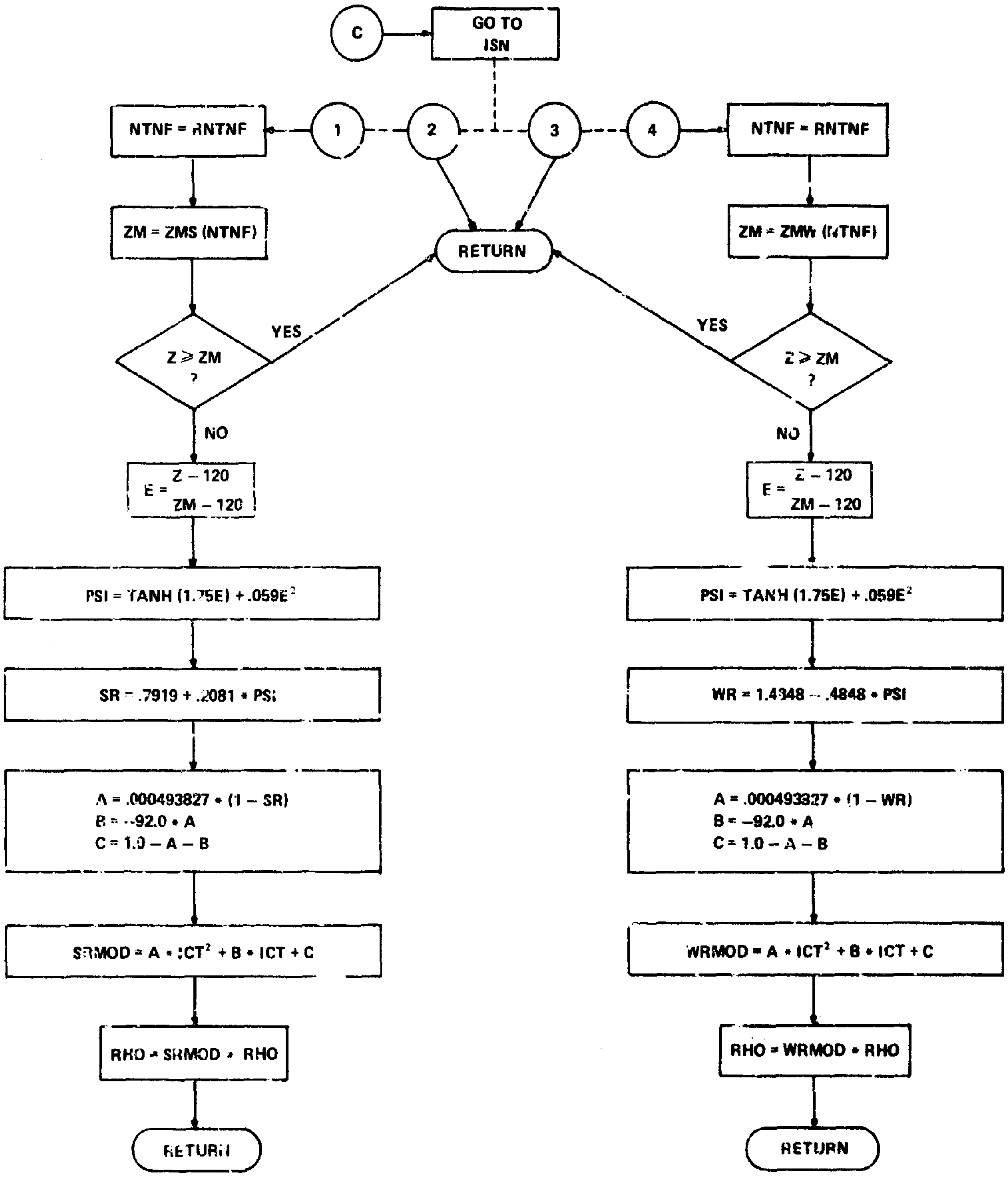


Figure IA5 Flowchart for Subroutine Density, Cont.

IA6. DIGITAL LISTING OF SUBROUTINE DENSTY

```
      SUBROUTINE DENSTY(R,LAMN,LN,RHO,IBNOUT,D,D0,GMT0,SFM,SFD,TNFDLT,
     1 ICT,ISN,T)
      REAL M,N,LN,LAMN,LSTA
      REAL LOGN(4)
      INTEGER GMT0,ZMS(11),ZMW(11),D,D0
      DIMENSION DA(3)
      COMMON /CONSTS/ G0,R0,BRNOUT
      DATA PI/3.141593/
      DATA AR,M,N /0.28,1.5,2.5/
      DATA (ZMS(J),J=1,11)/195,200,210,220,230,235,245,250,255,255,255/
      DATA (ZMW(J),J=1,11)/220,225,230,235,240,240,245,250,255,255,255/
      DATA BETA,P,GAMMA /-0.785398,0.20944,0.785398/
      E30 = 1.0E30
C Z IS ALTITUDE IN KILOMETERS
      Z = .001*(R-R0)
      IF (Z.LT.BRNOUT) GO TO 30
      IPD = D
      ID = D0 + T/86400.0
      D = MOD(ID,365)
      IF (D-IPD.EQ.1) ICT = ICT + 1
      F = (0.37+0.14*SIN(2.0*PI*(D-151)/365))*SIN(4.0*PI*(D-59)/365)
      TNF0 = 362.0 + 1.8*SFD + (1.8+F)*SFM
      GMT = GMT0 + T/36.0
      LSTA = GMT + LAMN*2400.0/(2.0*PI)
      LST = AMOD(LSTA,2400.0) + 0.5
      ETA = 0.5*ABS(LN)
      THETA = ETA
C H IS H*
      H = ((LST-1200)/1200)*PI
      TAU = H + BETA + P*SIN(H+GAMMA)
      IF (TAU.GT.PI) TAU = TAU -2.0*PI
      IF (TAU.LT.-PI) TAU = TAU + 2.0*PI
      A = AR*(((COS(ETA))**M - (SIN(THETA))**M)/(1+AR*(SIN(THETA))**M))
      RATIO = (1+A*(COS(TAU/2.0))**N)*(1+AR*(SIN(THETA))**M)
      TNF = TNF0*RATIO
      TNF = TNF + TNFDLT
      IF (TNF.LT.600.0) TNF = 600.0
      IF (TNF.GT.2100.0) TNF = 2100.0
      IF (TNF.GT.1100.0) GO TO 5
      RNTNF = (TNF-500.0)/100.0
      GO TO 6
    5 RNTNF = (TNF+100.0)/200.0
    6 CONTINUE
      TM = TNF - 800.0
      Q = TM/(750.0 + .0001722*TM*TM)
      S = .8730*EXP(-Q*Q/2.0)
      DA(1) = -S
      DA(2) = TNF/2500.0
      DA(3) = Z/1000.0
      CALL ODAJMO(0,3,DA,IERR,JCHAN)
ASSEMBL
        TSL      ='6041,,2     RESET SENSE LINE 1
        SFL      ='6041,,2    RESET CL1  OPERATE
STU     TSL      ='6041,,2    TEST SL1
```

```
       JNZ      STU
FORTRAN
      CALL QADCVO(0,4,LOGN,IERR,JCHAN)
/       SFL     ='6061,,2     SET CL1  IC
      RHO = 4.6515E-35*(E30)**LOGN(1)'+ 2.6561E-35*(E30)**LOGN(2)
     1 + 5.3122E-35*(E30)**LOGN(3) + 0.66453E-35*(E30)**LOGN(4)
C SEASON             SUMMER      FALL       WINTER     SPRING
C      ISN              1           2           3           4
C ICT IS HOW MANY DAYS INTO THE SEASON
      IF (ICT.LT.92) GO TO 9
      ISN = MOD(ISN,4) + 1
      ICT = 1
    9 GO TO (70,71,72,73),ISN
   70 NTNF = RNTNF
      ZM = ZMS(NTNF)
      IF (Z.GE.ZM) GO TO 73
      E = (Z-120.0)/(ZM-120.0)
      PSI = TANH(1.75*E) + 0.059*E*E
      SR = 0.7919 + 0.2081*PSI
      A = .000493827*(1.0-SR)
      B = -92.0*A
      C = 1.0-A-B
      SRMOD = A*ICT*ICT + B*ICT + C
      RHO = SRMOD*RHO
      RETURN
   71 CONTINUE
   73 RETURN
   72 NTNF = RNTNF
      ZM = ZMW(NTNF)
      IF (Z.GE.ZM) GO TO 73
      E = (Z-120.0)/(ZM-120.0)
      PSI = TANH(1.75*E) + 0.059*E*E
      WR = 1.4848 - 0.4848*PSI
      A = .000493827*(1.0-WR)
      B = -92.0*A
      C = 1.0-A-B
      WRMOD = A*ICT*ICT + B*ICT + C
      RHO = WRMOD*RHO
      RETURN
   30 IBNOUT = 1
      RETURN
      END
```

IB. OPERATING INSTRUCTIONS FOR SIMSAT II

The orbiting satellite simulation program with the hybrid air density model has been given the name Simsat II.

For the sake of completeness, the operating instructions, although an independent self-contained document, are included in this section.

OPERATING INSTRUCTIONS

FOR

SIMSAT II

(Satellite Trajectory Simulation Program)

Prepared by

Stuart B. Mindlin

ELECTRONIC ASSOCIATES, INC.
Princeton Computation Center
Princeton, New Jersey 08540

Prepared for

AIR FORCE CAMBRIDGE RESEARCH LABS.
United States Air Force
Bedford, Massachusetts 01730

October 8, 1969

SIMSAT II is a satellite trajectory simulation program written for the 8900 hybrid computer. This program was designed for use in evaluating the effects of various perturbative forces, such as atmospheric drag, solar radiation pressure, and gravitational anomalies, upon the trajectory and lifetime of an orbiting satellite. SIMSAT II is identical to SIMSAT I with the addition of a more sophisticated density model as discussed in U. S. Standard Atmosphere Supplements, 1966. Integration of equations to calculate atmospheric density and, in turn, drag force is done on the 8800 analog portion of the hybrid system. In addition variables of interest are transferred to the 8800 analog computer where they may be observed and plotted. To operate SIMSAT II, the following steps must be followed.

1) Patch a pair of 8800 analog and logic panels as shown in Figures IA1, IA2, and IA3. These are the equations for molecular density of the major atmospheric constituents. The additional patching allows the user to plot latitude versus longitude on the x-y plotter and also to control the transfer of program variables to the analog computer.

2) Punch a set of data cards which specify initial position and velocity of the satellite, and the scale factors to be used in digital/analog transfers, as well as initial data needed for the density calculation. (See Table 2, section 7.)

3) Mount the 8800 patch panels. Switch 1013 off.

4) Load and execute the digital program.

5) Control of the running program may be maintained from either the analog or digital consoles as follows:

   a) If console register switch 8 is on, control is from the digital console, otherwise it is from the analog console.

   b) If control is from the digital console (C8 on), then the mode of the simulation is determined by the following table.

Mode	Console Switch 1	Console Switch 2
IC	ON	OFF
IC	ON	ON
HOLD	OFF	OFF
OP	OFF	ON

   c) If it is desired to print out the values of the program variables on the 8400 line printer, turn on console register switch 7. All program variables will have their values printed out once, and the console switch will be reset. This may be done whether the simulation is under

analog or digital control, but only when in OP or HOLD modes.

d) If it is desired to read in new initial conditions for the satellite, this may be done by pressing console register switch 10, when the simulation is in IC mode, under either analog or digital control.

e) If console register switch 9 is set, the program will go into HOLD mode after each complete orbit of the satellite. This is useful in checking out the change in orbital parameters from one orbit to the next.

f) Which program variables are transferred to the analog computer is controlled by four logic function switches on the 8800 console, as follows:

DAC CHANNEL	FUNCTION SWITCH 011 LEFT OR CENTER		FUNCTION SWITCH 011 RIGHT
3	radius		radial perturbation
4	longitude in orbital plane		perturbation in orbital longitude
5	velocity normal to radius		perturbation in tangential velocity
6	velocity in radial direction		perturbation in radial velocity
7	geographical longitude		Keplerian radius
8	geographical latitude		Keplerian orbital longitude
9	altitude		Keplerian tangential velocity
10	air density		Keplerian radial velocity
11	x component of drag force		x component of drag force
12	y component of drag force		y component of drag force
13	x component of velocity	See	tangential velocity
14	y component of velocity	Table	zero
15	z component of velocity	1	radial velocity

As outlined in the report AFCRL 69-0123 several coordinate frames are used in the simulation. The satellite's velocity is resolved into each of these coordinate systems, and the x,y, and z components of any of these velocity representations may be transferred on DAC channels 13, 14, and 15, respectively. Which velocity components are transferred is

controlled by logic switches 1012, 1013. and 1211 as follows:

TABLE 1

Switch 1012	Switch 1013	Switch 1211	Velocity Transferred
OFF	OFF	OFF	H frame components
OFF	OFF	ON	H frame components
OFF	ON	OFF	Modified Euler frame components
OFF	ON	ON	Orbital frame components
ON	OFF	OFF	Inertial frame components
ON	OFF	ON	Navigational frame components
ON	ON	OFF	Euler frame components
ON	ON	ON	Euler frame components

7) The format of the data cards required by SIMSAT II is described below.

The first four cards contain the scale factors to be used in the D/A transfer of program variables. Cards 1 and 2 contain the scale factors to be used when logic switch 1012 is on, and cards 3 and 4 contain the scale factors to be used when switch 1012 is off. Each scale factor is punched in 10 column floating point format. The fifth card contains g_O, the gravitational constant in columns 1-10, R_O, the radius of the earth, in columns 11-20, and H_{MIN}, the altitude below which the satellite is considered to have burnt out, in columns 21-30. During execution of SIMSAT II, if the altitude of the satellite goes below H_{MIN}, the run is terminated and the time at which burnout occurred is printed out. The last three values on the fifth card are for use in the density calculation. The 3-hour geomagnetic planetary index, K_p, is found in columns 31-40. Columns 41-50 and 51-60 contain respectively the monthly and daily means of the 10.7 cm. solar flux. (See U. S. Standard Atmosphere Supplements, 1966, pg. 47.) Card number 6 contains either "YES" or "NO" in columns 1-3, to indicate if the satellite's initial position and velocity, orbital insertion time and day are to be read in from cards, or typed in on the console typewriter. NO means the data

is to be read in from cards, in which case three more data cards are required which contain the following initial data, the first two of which are in 10 column floating point format: Radius in meters, longitude in degrees, latitude in degrees, velocity in meters/sec., azimuth in degrees, flight path angle in degrees, mass in kg., area in square meters, drag coefficient, and time scale factor (factor by which simulation is to be sped up over real time, normally (1000.0). The third card contains the day of the year in columns 5-7 and Greenwich Mean Time at orbit insertion in columns 11-14. The layout of all the data cards is summarized in Table 2.

TABLE 2

Card	Columns	Scale Factors Should be the Reciprocals of the Maximum Valve
1	1-10	Scale factor for DAC3, switch 011 right.
1	11-20	Scale factor for DAC4, switch 011 right.
1	21-30	Scale factor for DAC5, switch 011 right.
1	31-40	Scale factor for DAC6, switch 011 right.
1	41-50	Scale factor for DAC7, switch 011 right.
1	51-60	Scale factor for DAC8, switch 011 right.
1	61-70	Scale factor for DAC9, switch 011 right.
1	71-80	Scale factor for DAC10, switch 011 right.
2	1-10	Scale factor for DAC11, switch 011 right.
2	11-20	Scale factor for DAC12, switch 011 right.
2	21-30	Scale factor for DAC13, switch 011 right.
2	31-40	Scale factor for DAC14, switch 011 right.
2	41-50	Scale factor for DAC15, switch 011 right.
2	51-60	(Blank)
2	61-70	(Blank)
2	71-80	(Blank)
3	1-10	Scale factor for DAC3, switch 011 left or center.
3	11-20	Scale factor for DAC4, switch 011 left or center.
3	21-30	Scale factor for DAC5, switch 011 left or center.
3	31-40	Scale factor for DAC6, switch 011 left or center.
3	41-50	Scale factor for DAC7, switch 011 left or center.
3	51-60	Scale factor for DAC8, switch 011 left or center.

TABLE 2 (continued)

Card	Columns	Data
3	61-70	Scale factor for DAC9, switch 011 left or center.
3	71-80	Scale factor for DAC10, switch 011 left or center.
4	1-10	Scale factor for DAC11, switch 011 left or center.
4	11-20	Scale factor for DAC12, switch 011 left or center.
4	21-30	Scale factor for DAC13, switch 011 left or center.
4	31-40	Scale factor for DAC14, switch 011 left or center.
4	41-50	Scale factor for DAC15, switch 011 left or center.
4	51-60	(Blank)
4	61-70	(Blank)
4	71-80	(Blank)
5	1-10	g_o, gravitational const. (m/sec.^{2}).
5	11-20	R_o, radius of earth (meters).
5	21-30	H_{MIN}, burnout altitude (kilometers).
5	31-40	K_p, geomagnetic planetary index.
5	41-50	Monthly mean of 10.7 cm solar flux.
5	51-60	Daily mean of 10.7 cm solar flux.
6	1-3	"YES" or "NO", input is/is not from typewriter.
7	1-10	Initial radius (meters).
7	11-20	Initial longitude (degrees east of Greenwich).
7	21-30	Initial latitude (degrees north of equator).
7	31-40	Initial velocity (meters/sec.).
7	41-50	Initial azimuth (degrees clockwise from north).
7	51-60	Initial flight path angle (degrees away from earth).
7	61-70	Satellite mass (kilograms).
7	71-80	Satellite cross-sectional area (square meters).
8	1-10	Drag coefficient.
8	11-20	Time scale factor.
9	5-7	Day of the year.
9	11-14	Greenwich Mean Time.

If card no. 6 is YES, indicating typewriter input, the user should follow the directions typed out, and input the initial data as it is requested.

8. After new initial conditions are read the system will type that the operator may set a certain console switch for a density check.

If the operator sets this switch the system will automatically perform a dynamic check of the entire hybrid air density model. The results of the check is output on the lineprinter in the form of the tables of SPRING-FALL density in the U. S. Standard Atmosphere Supplements, 1966.

Comparison of these values will confirm proper operation of the model, or if there are difficulties, help locate the source.

II. PROPAGATION OF ELECTROMAGNETIC ENERGY IN THE IONOSPHERE

Preceding page blank

II.1.0 INTRODUCTION

This section of the report describes modifications which have been made to the basic hybrid ray tracing program that was developed under a previous contract (No. F 19628-67-C-0359). The results and derivations of the ray tracing equations were included in the final report on that contract: AFCRL Report 69-0123, "Hybrid Computer Applications to Mathematical Models of Physical Systems," February 1969.

The modifications to the program were as follows:

(1) Addition of electron collisions.

(2) Addition of an automatic covergence scheme based on the range between transmitter and receiver.

(3) Time history storage of pertinent variables for the optimum ray path when determined by the covergence scheme.

II.2.0 DESCRIPTION OF ELECTRON COLLISIONS

The ray tracing equations solved to determine the path of electromagnetic energy transmitted through the ionosphere are described in AFCRL Report 69-0123, Section III. In these equations the expression for the refractive index neglected the effect of electron collisions. Under the current contract the Appleton-Hartree formula for the refractive index was modified to include the general case of electron collisions. This formula in general form is:

$$\mu^2 = 1 - \frac{x}{1 - iz - \frac{Y_t^2}{2(1-x-iz)} \pm \left\{ \frac{Y_t^4}{4(1-x-iz)^2} + Y_L^2 \right\}^{\frac{1}{2}}}$$

where,

μ = refractive index

$x = \frac{f_N^2}{f^2}$

$f_N = 8.98 \times 10^3 \times N^{\frac{1}{2}}$, the plasma frequency

f = wave frequency

N = electron density (electrons/cc)

Y = normalized magnitude of the earth's magnetic field vector

$= \left| \frac{1}{2\pi f} \quad \frac{eB}{m} \right|$

ψ = angle between $\vec{Y}$, earth magnetic field vector, and the wave normal

$Y_t = Y \sin \psi$

$Y_L = Y \cos \psi$

z = electron collision frequency

$+ \Rightarrow$ an ordinary ray

$- \Rightarrow$ an extra ordinary ray

II.3.0 DESCRIPTION OF CONVERGENCE SCHEME

The convergence scheme used is basically a gradient method which corrects the firing angle (α) of the transmitter based on the longitude and altitude of the ray path upon arrival at the latitude of AFCRL. At the outset it was thought desirable to correct not only α, but also the azimuth angle (β). However, it was later decided to maintain β constant since magnetic field effects were not thought to be of significant magnitude as to cause large deviation in β along the path of the ray.

By way of introduction to the scheme used, it should first be pointed out that for a given electron density distribution, the approximate range of firing angle is not in general known. For this reason, it was thought best to choose three firing angles separated sufficiently far apart so as to determine the approximate area in which to advance (see flow chart and convergence scheme flow for specifics of program).

The first three paths for a given electron density profile are shown qualitatively below. In this instance, the path which is closest to the desired path is curve number 3.

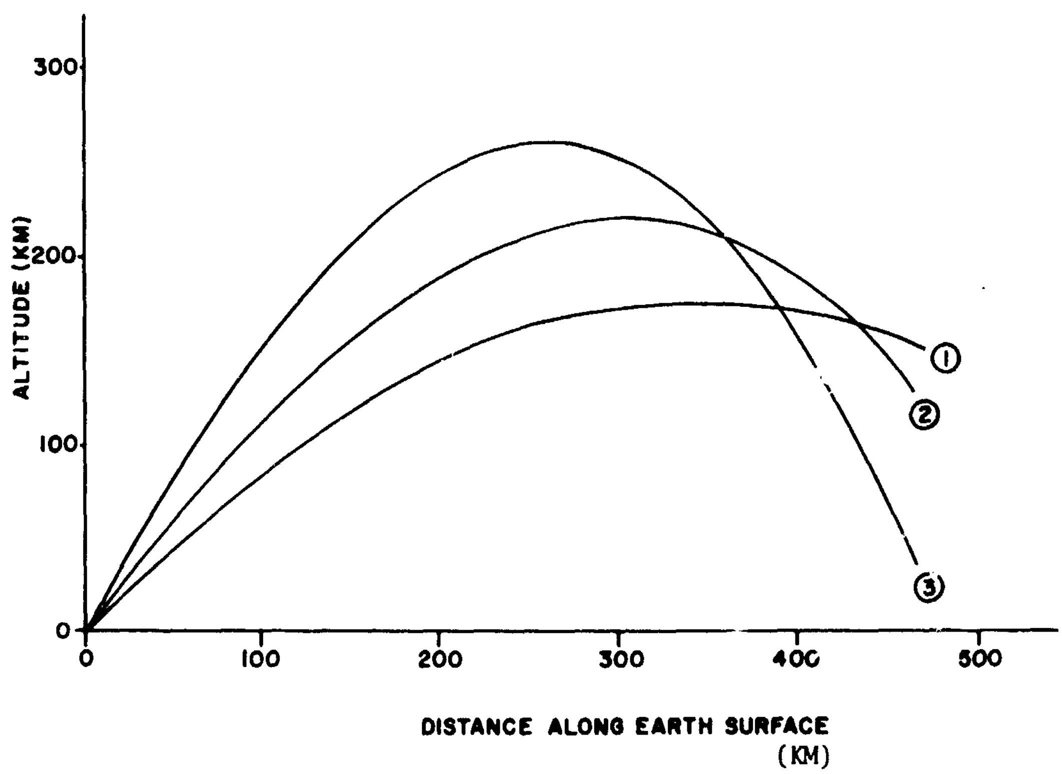


FIGURE (II-1)

II.3.1 CONVERGENCE SCHEME MATHEMATICS

If α^*, β^* are defined to be the firing and azimuth angle, respectively, which gave the minimum altitude at AFCRL latitude, we write

$$(1) \quad D(\alpha^*, \beta^*) = H(\alpha^*, \beta^*) + A\,(\theta - \theta_T)$$

where A is a constant to be defined which will dictate the necessary tolerance on θ (longitude), $H(\alpha^*, \beta^*)$ is the final height, and θ_T = longitude (target) of AFCRL.

D is the quantity which is monitored for the optimum ray path, i.e. if we require a 1 km tolerance on the final altitude $H(\alpha, \beta)$ and a .1 tolerance on $(\theta - \theta_T)$ then equation (1) becomes

$$(2) \quad D(\alpha^*, \beta^*) = 1 + 1(.1) = 1.1$$

so that when a ray path is found whose final conditions satisfy (2), then this is the optimum ray. As it turns out magnetic field deflections are slight and as a result the boundary condition of equation 2) is effectively a function of final altitude only, i.e. stopping the hybrid program at the latitude of AFCRL also means stopping at the correct longitude.

Of course, in general one of the first three rays will give us only an approximation of the general area in which to advance. Therefore, since equation (1), (2) will most often not be satisfied, the automatic gradient scheme takes over to choose a new alpha and beta increment as follows: To start the process α and β are incremented (for the general case; only α was incremented, β was held constant) by a constant (1°) to determine to slope. First increment alpha and monitor the value of D. We then have

$$(3) \quad D(\alpha^*+\Delta\alpha, \beta^*) = H(\alpha^*+\Delta\alpha, \beta^*) + A\,(\theta(\alpha^*+\Delta\alpha, \beta^*) - \theta_T)$$

and as an approximation,

$$(4)\quad \frac{\partial D}{\partial \alpha} = \frac{D(\alpha^* + \Delta\alpha, \beta^*) - D(\alpha^*, \beta^*)}{\Delta\alpha}$$

Now increment β and write,

$$(5)\quad D(\alpha^*, \beta^* + \Delta\beta) = H(\alpha^*, \beta^* + \Delta\beta) + A(\theta(\alpha^*, \beta^* + \Delta\beta) - \theta_T)$$

and,

$$(6)\quad \frac{\partial D}{\partial \beta} = \frac{D(\alpha^*, \beta^* + \Delta\beta) - D(\alpha^*, \beta^*)}{\Delta\beta}$$

using this information we must now choose the next α, β pair

The magntidue of the gradient of D is defined as

$$(7)\quad (\nabla D)^2 = \frac{\partial D}{\partial \alpha}^2 + \frac{\partial D}{\partial \beta}^2$$

or

$$(8)\quad \nabla D = \sqrt{\frac{\partial D}{\partial \alpha}^2 + \frac{\partial D}{\partial \beta}^2}$$

Therefore we may write

$$(9)\quad \Delta\alpha_{NEW} = \frac{H(\alpha^* + \Delta\alpha, \beta^*)}{(\nabla D)^2} \frac{\partial D}{\partial \alpha}$$

and,

$$(10)\quad \Delta\beta_{NEW} = \frac{(H(\alpha^*, \beta^* + \Delta\beta))}{(\nabla D)^2} \frac{\partial D}{\partial \beta}$$

which reduces to (when $\frac{\partial D}{\partial \beta} = 0$),

$$(11)\quad \Delta\alpha_{NEW} = \frac{H(\alpha^* + \Delta\alpha, \beta^*)}{\frac{\partial D}{\partial \alpha}}$$

and

$$(12) \quad \Delta\beta_{NEW} = 0$$

Using the $\Delta\alpha$ the hybrid program starts anew and the same process is repeated until convergence criteria (equations 1, 2) are met.

Once the optimum ray path is reached it is repeated so that the time history array may be stored and printed at the end. This is the end of one run for a given electron density profile.

II.3.2 CONVERGENCE SCHEME FLOW

As mentioned in the introductory remarks to this section, the first phase of the convergence scheme flow is to pick three firing angles and determine the ray path for each. These angles as may be seen from the flowchart Apendix A3 are 12.5, 25, 37.5 degrees. Referring to the system flowchart, at statement 615 we begin the scheme. After first having chosen α, set the necessary pots, and initialized the program, we enter the hybrid loop at statement number 7Ø. TOL1 is the Fortran constant which specifies the tolerance on the latitude as a stopping condition, i.e. once DIFF, which is the difference between the present latitude and the final latitude, is less than or equal to TOL1 we fall out of the hybrid loop and call subroutine QHOLDØ. This subroutine (which is part of the hybrid run time library) places the analog computer in the HOLD mode.

The next decision block is necessary for the time history storage END is a logical variable which is set true upon completion of the run which satifies the criterion $D(\alpha^*, \beta^*) \leq 1.1$ as expressed by equation (2) of section (3.1). Once END becomes true we repeat

the run so that we may store the pertinent time history variables. The criterion expressed above is stated in the Fortran program as DIFFD $\leq$ TOL2; hence the decision block for DIFFD.

If DIFFD is greater than TOL 2 and the first three firing angles have been completed, we increment α (statement number 651) and proceed through the program again. The remainder of the flowchart follows exactly the mathematical description in the last section. This procedure begins at statement number 17Ø1. The following equivalences are helpful in comparing the FORTRAN program to the mathematics:

$$\text{DADA} = D(\alpha^*, \beta^*)$$
$$\text{PDPA} = \frac{\partial D}{\partial \alpha}$$
$$\text{PDPB} = \frac{\partial D}{\partial \beta}$$
$$\text{GRADD2} = \nabla D^2 = \frac{\partial D}{\partial \alpha}^2 + \frac{\partial D}{\partial \beta}^2$$
$$\text{GRADD} = \nabla D$$
$$\text{DALPH} = \Delta\alpha$$
$$\text{DBETA} = \Delta\beta$$
$$\text{AORIG} = \alpha \text{ original}$$
$$\text{BORIG} = \beta \text{ original}$$

Note the presence of the logical variable NOTBE on the third page of the flowchart. This variable acts as an indicator in determining if β is to be corrected along with α. As mentioned previously, β optimization was not desired, thus NOTBE was false during all runs.

II.4.0 RESULTS

If the ray paths are not convergent the automatic scheme can do nothing. In this case there is no time history stored. In running the various twelve electron density profiles eight of

twelve profiles would not allow convergence to occur at the given altitude, latitude and longitude of AFCRL. Essentially what happens is that the program chooses the next angle based on the previous calculation and finds that instead of decreasing the final altitude it increases it, i.e. the final altitude as a function of firing angle α reaches, at least, a local minimum. This event is described qualitatively by Figure (II-2).

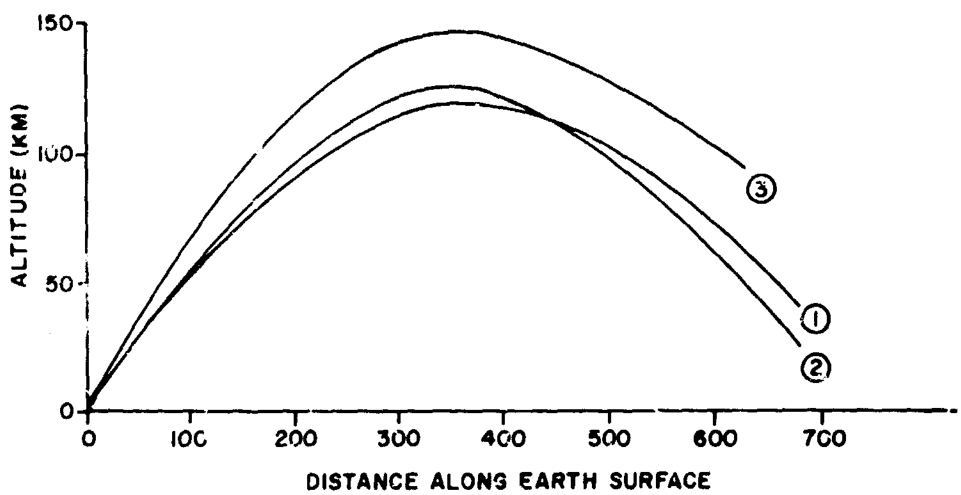


FIGURE (II-2)

In this case the program tries again to decrease the height by going in the other direction and this process becomes endless. On most of the rays which did not converge the final altitude was in the range of 100-150 km.

These results were verified by the conventional ray tracing program with no convergence scheme.

Based on the fact that the automatic scheme showed non-convergence for eight of twelve electron density profiles it was desired by AFCRL to have the running procedure for the conventional hybrid ray tracing program. The procedures supplied for both programs are included in Appendix II.

II.5.0 CONCLUSIONS AND RECOMMENDATIONS

Based on the results of the convergence scheme, it might be desirable to first make a small modification to the automatic program which would allow the program to pick say the ray path which gave the absolute minimum altitude at the latitude and longitude of AFCRL, for a given profile. Once this is completed, it would be informative to introduce perturbations into the electron density distribution during a transmission for the ray path determined above. The results of this might be used for comparison with ionosonde data. For example what perturbations (or random disturbance pattern) introduced into the program correlate most nearly with ionosonde data regarding intensity, dispersion etc.

The results of study of this nature might be very useful in predicting at least qualitative changes in electron density distributions.

APPENDIX II

IIA1. ANALOG SCHEMATICS

Figures (IIA1.) and (IIA2.) are the analog diagrams of the automatic system.

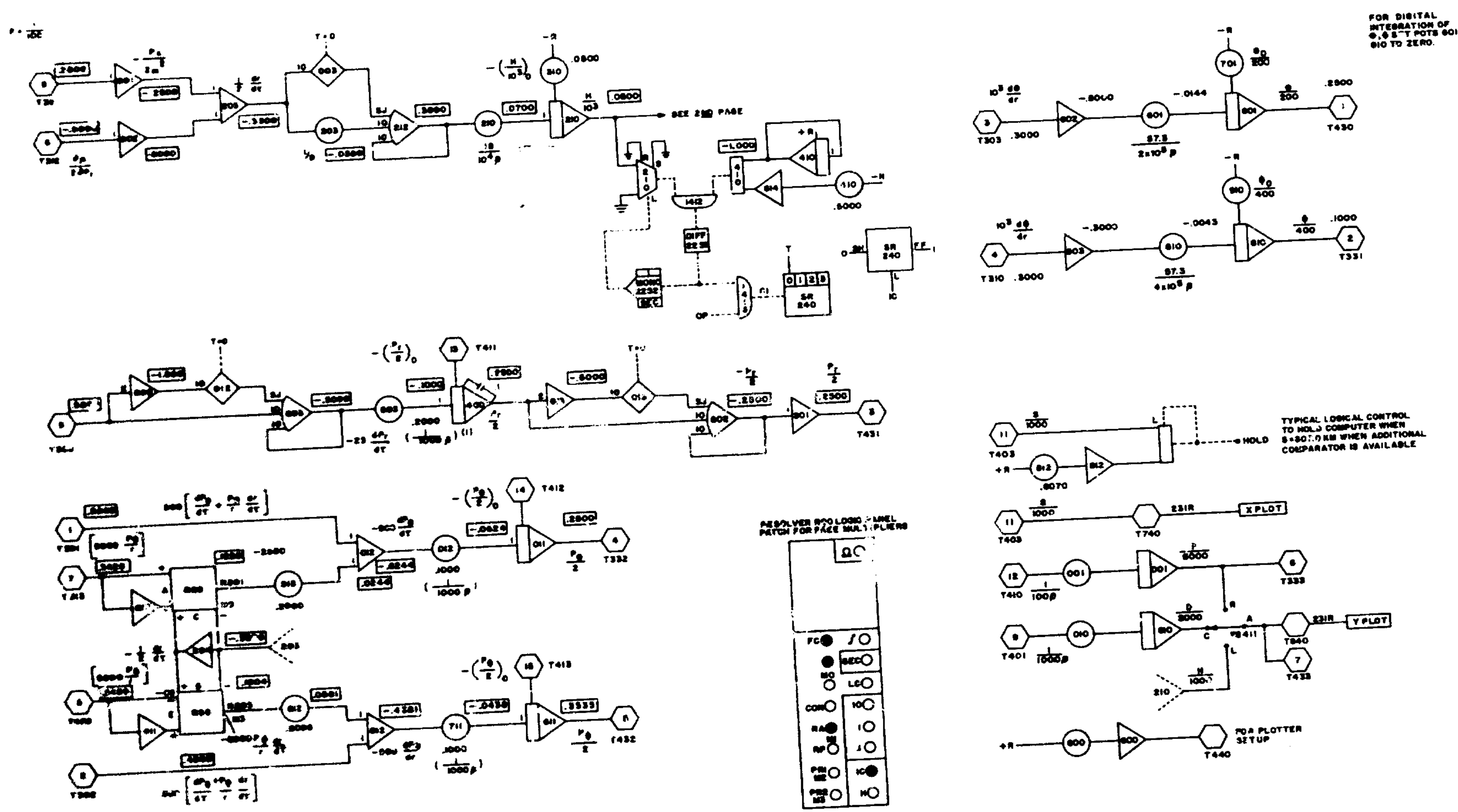


Figure (IIA1.) Main Analog Program

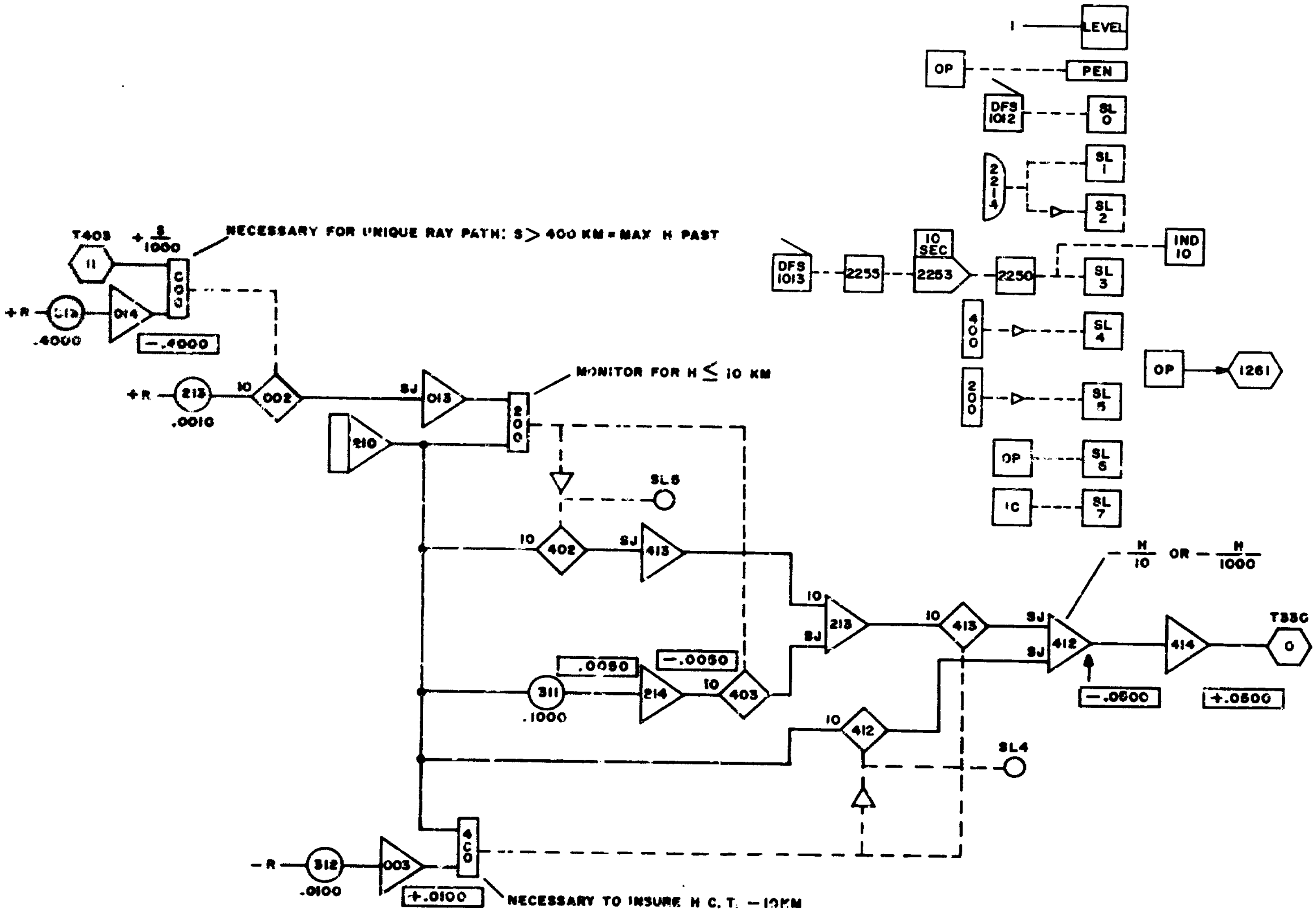


Figure (IIA2). AUTOMATIC SEARCH ROUTINE FOR OPTIMUM H

IIA2. COMPONENT ASSIGNMENTS

Pages 66 through 69 are the pot and amp. assignment sheets plus static test voltages. Page 70 is the switch assignment sheet for the sense lines on the 8800.

Pot Number	Parameter	Setting
C203	Scaling	.1111
C210	$\frac{18}{10^4 \beta}$	.1800
C310	$\frac{Ho}{10^3}$ (IC)	.0500
C410	Scaling	.5000
C601	$\frac{57.3}{2x10^5 \beta}$	.0286
C810	$\frac{57.3}{4x10^5 \beta}$	.0143
C812	$\frac{S_{Final}}{1000}$	.6070
C001	$\frac{1}{100 \beta}$	1.0000
C010	$\frac{1}{1000 \beta}$	.1000
C603	$\frac{2}{1000 \beta}$	.2000
C513	Scaling	.2000
C612	Scaling	.2000
C012	$\frac{1}{1000 \beta}$	.1000
C711	$\frac{1}{1000 \beta}$	.1000
C313	Scaling	.4000

Pot Number	Parameter	Setting
C213	Scaling	.0010
C312	*Scaling for ht $\frac{H_f}{1000}$	.0100
C311	Scaling	.1000
C701	$\frac{\theta o}{200}$	.2500
C910	$\frac{\Phi o}{400}$	.1000

*H_f is the height at which there is a scale change on H for better resolution H_f = 10 km.

STATIC CHECK

Component	Value
A002	0.6000
A201	-0.2500
A203	-0.3500
A212	0.3890
A210	0.0500
A410	-0.9999
A602	-0.5000
A803	-0.3000
A601	0.2500
A810	0.1000
A800	-0.9999
A603	-0.5000
A400	0.2500
A613	-0.5000
A802	-0.2500
A801	0.2500
A412	0.0500
A414	0.0500
A413	0.0000
A204	0.3500
A411	-0.5439
A412	-0.0500
A012	-0.6244
A612	-0.4381
A011	0.2500
A611	0.3535
A214	-0.0000
A014	-0.4000
P203	-0.0389
P210	0.0700
P310	-0.0500

STATIC CHECK

Component	Value
P601	-0.0144
P810	-0.0043
P701	-0.2500
P910	-0.1000
P603	-0.1000
P513	+0.0244
P612	+0.0381
P012	-0.0624
P711	-0.0438
P311	+0.4000
P213	+0.0010
P311	+0.0050
P312	+0.0100
T311	+0.2500
T312	-0.6000
T303	+0.5000
T310	+0.3000
T300	+0.5000
T301	+0.6000
T313	+0.3486
T400	+0.5439
T302	+0.4000

SWITCH ASSIGNMENT
(AUTOMATIC)

Sense Line	Exc	Function
0	SW 1012	Print Out Check & Time Hist.
*1	SW 1211(1)	$P\Phi$ = f (Pr, $P\theta$, μ)
*2	SW 1053 (0)	$P\theta$ = f(Pr, $P\Phi$, μ)
3	SW 1013	Not Used
4	Used as Monitor	(for H<-10km)
5	Used as Monitor	(for H≤10km: Scale Change)
*6	OP	Hold Simulated
*7	IC	IC Simulated

*Asterisks denote sense lines which are hard wired.

IIA3. SYSTEM FLOW DIAGRAM

Figure (IIA3) represents the flowchart of the system. The purpose of this diagram is to show the flow of the convergence scheme. For derivation of and insight into the ray tracing equations see AFCRL Report A69-0123, Contract No. F19 628-67-C0359, February 1969.

Included also is the digital listing of the program.

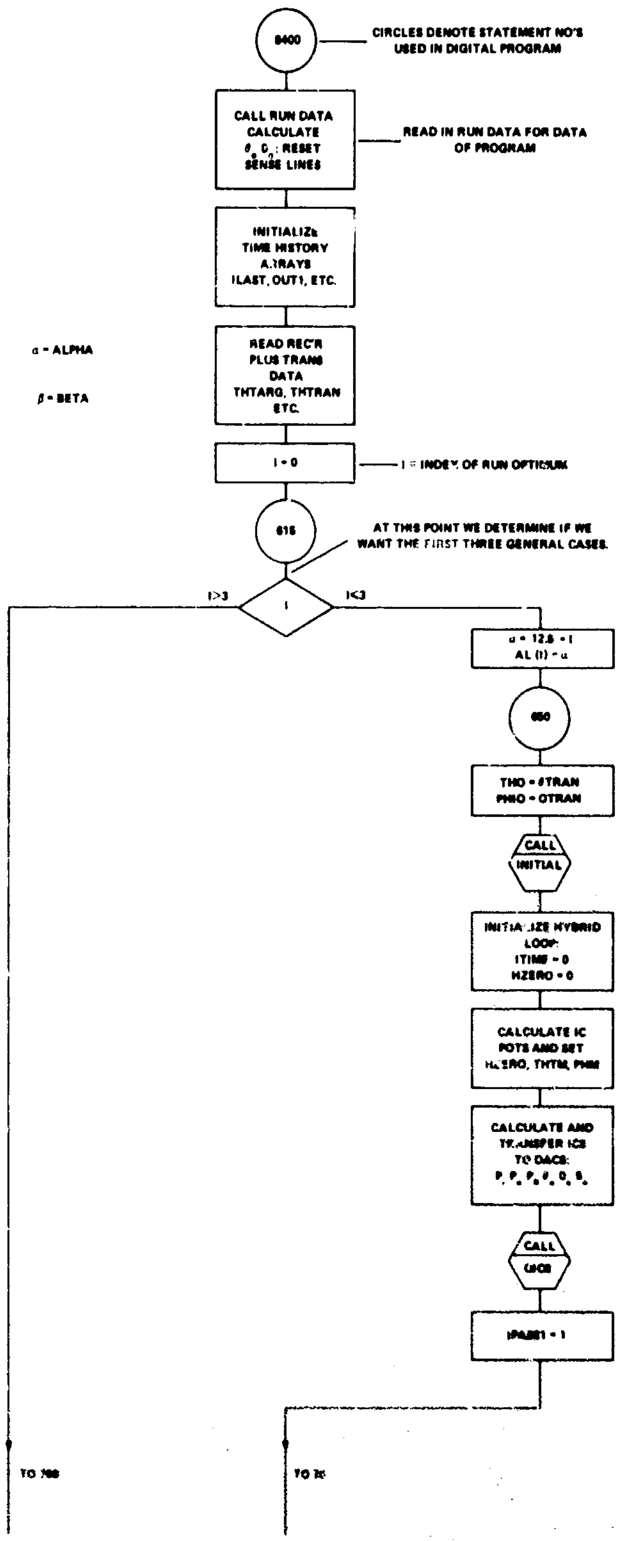


Figure (IIA3.) System Flowchart

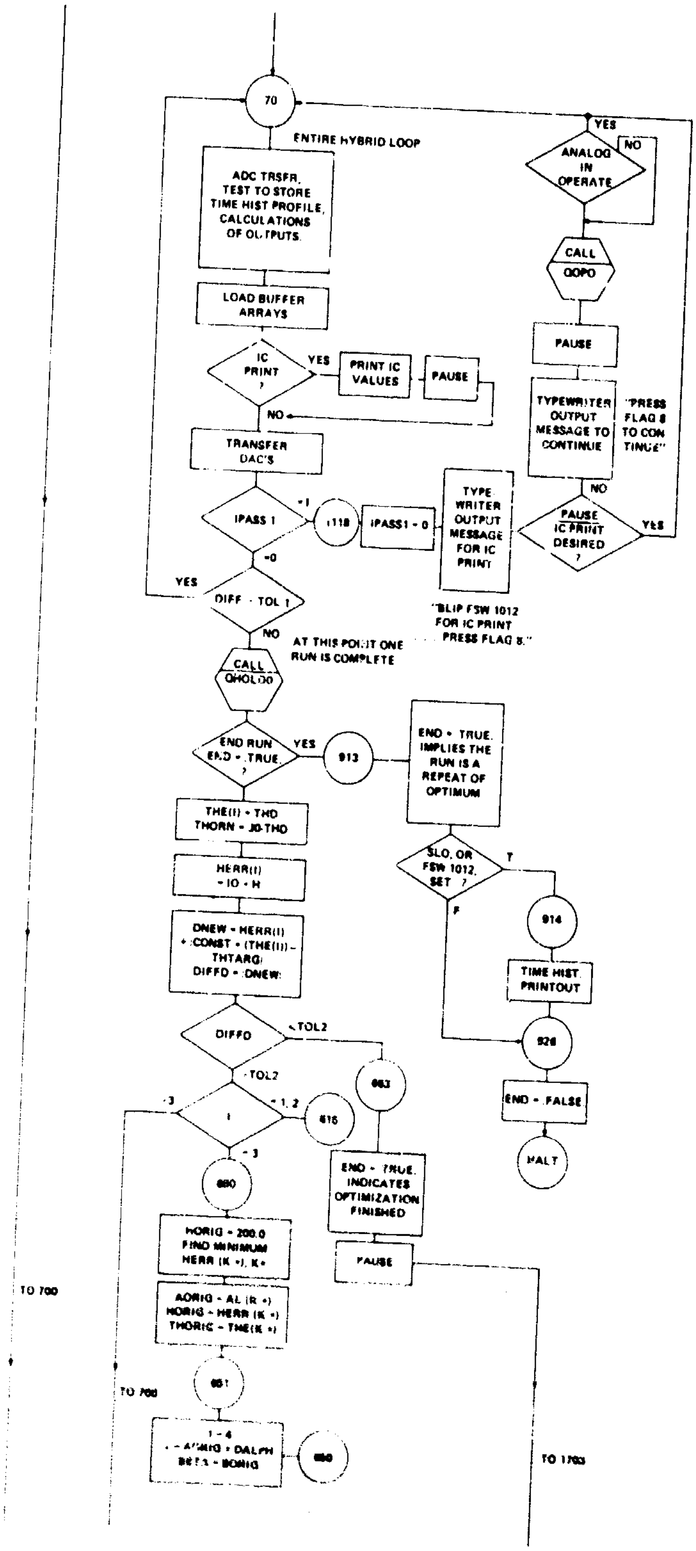


Figure (IIA3.) System Flowchart, Cont.

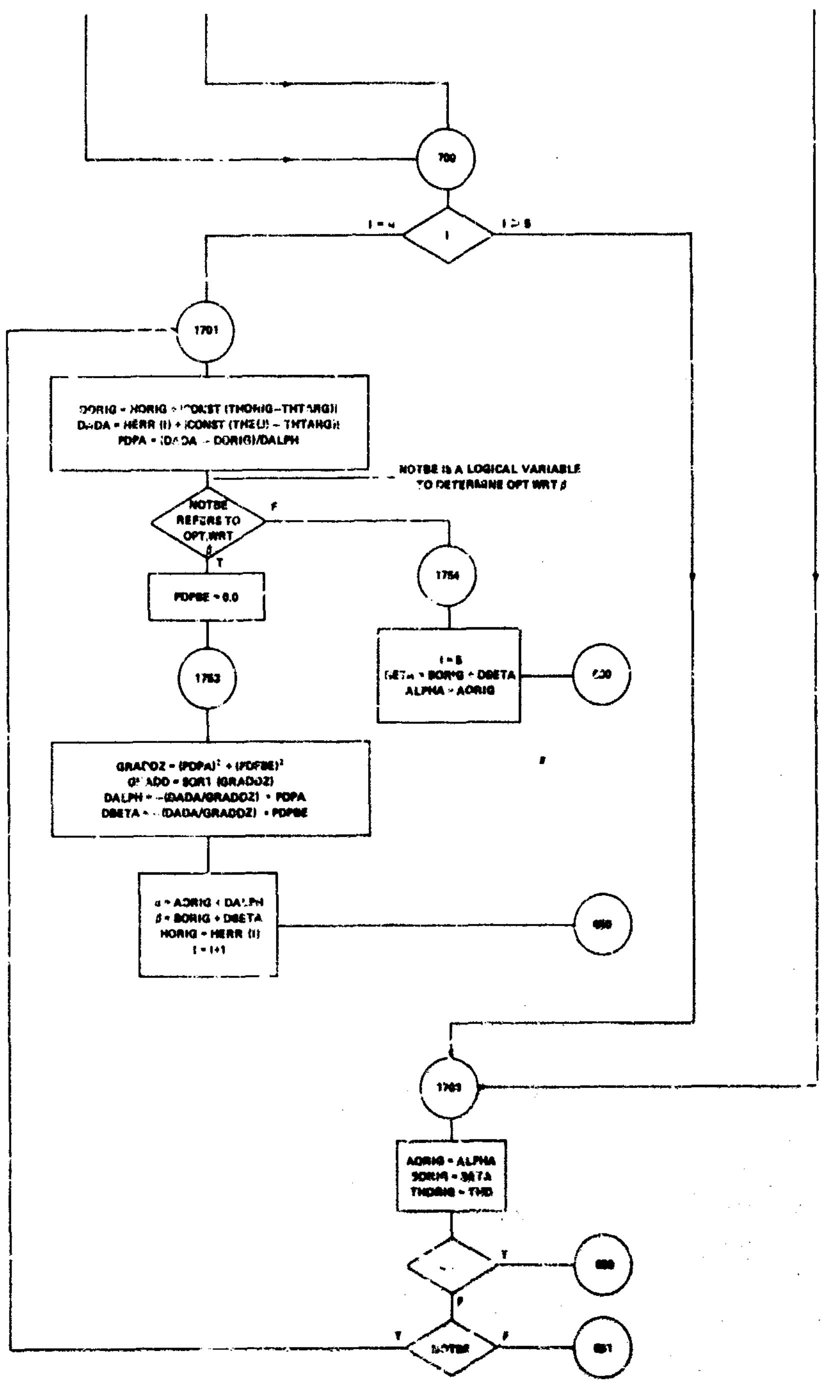


Figure (IIA3.) System Flowchart, Con:.

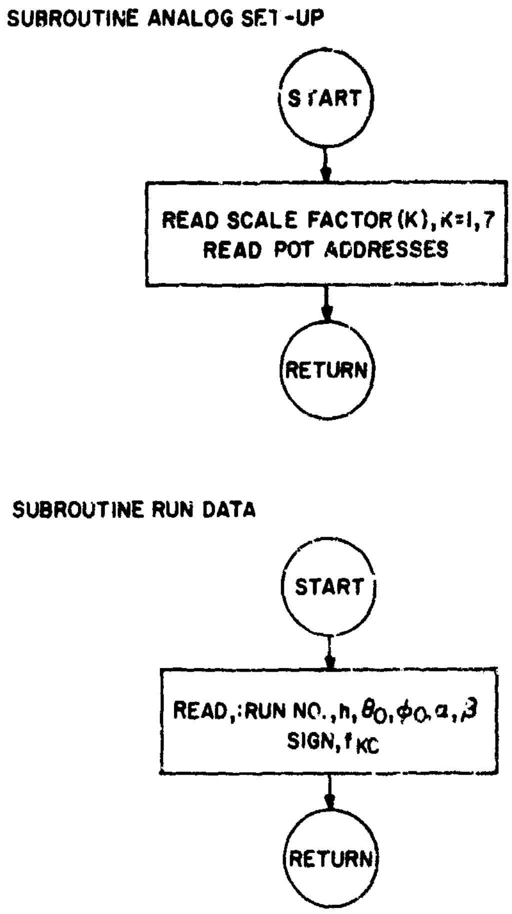


Figure (IIA4.) Subroutines Analog Set-Up and Run Data

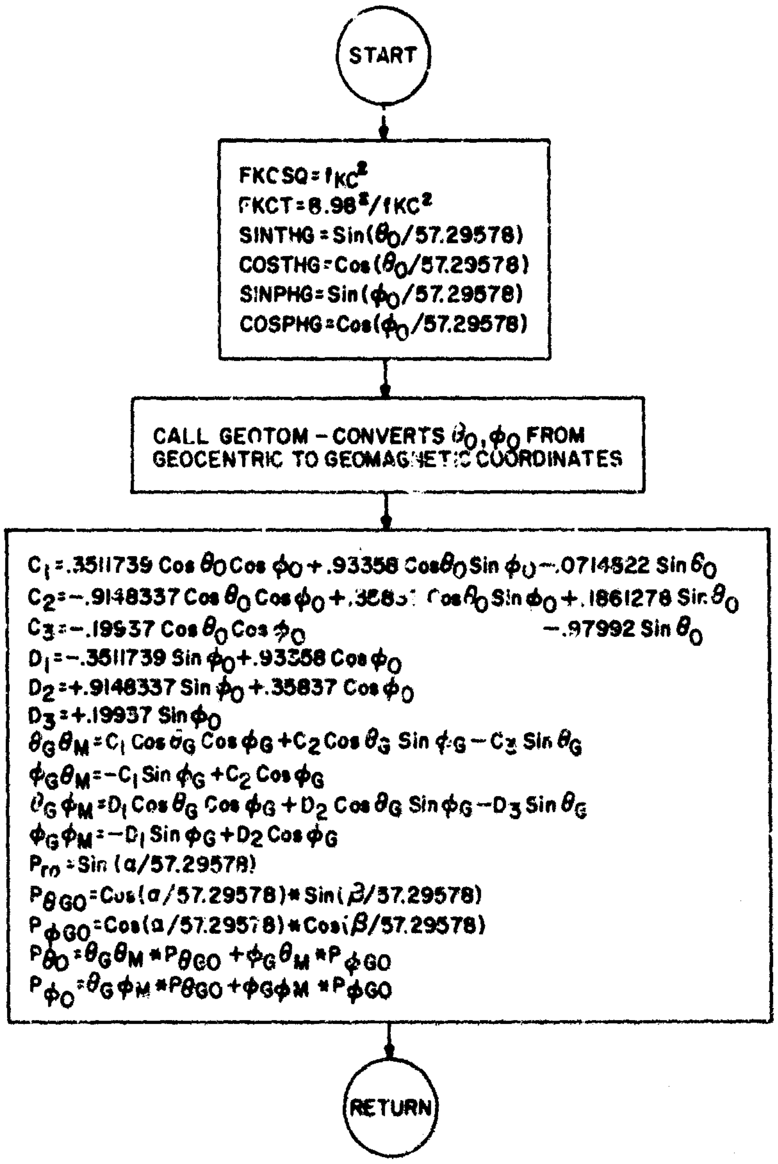


Figure (IIA5.) Subroutine Initial

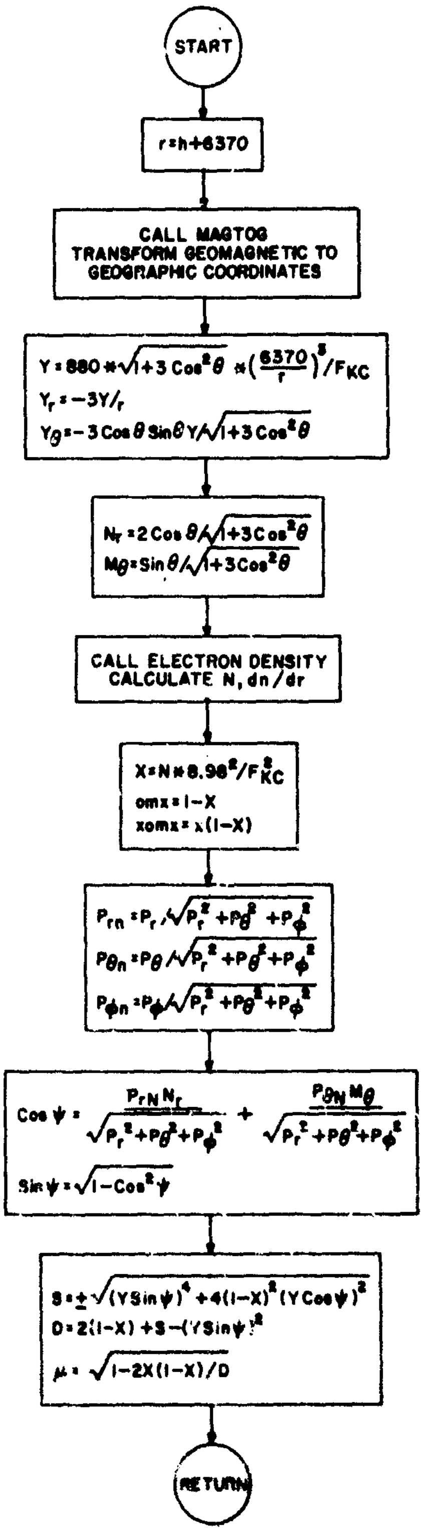


Figure (IIA6.) Subroutine Index of Refraction
REF: (PP(3-12) to (3-14) AFCRL 69-0123)

START

$$\frac{\partial\psi}{\partial\theta}=\frac{2}{1+3\,\mathrm{Cos}^2\theta}\left[\frac{\mathrm{Pr}M\theta-P\theta\mathrm{Nr}}{\mu\,\mathrm{Sin}\,\psi}\right]$$

$$D^2=\left[2(1-X)-Y^2\mathrm{Sin}^2\psi\pm\left[(Y\,\mathrm{Sin}\,\psi)^4+4Y^2(1-X)^2\mathrm{Cos}^2\psi\right]^{1/2}\right]^2$$

$$\mu^2\,\mathrm{Sin}\,\psi=\left(1-\frac{2X(1-X)}{D}\right)\mathrm{Sin}\,\psi$$

$$\frac{1}{\mu}\frac{\partial\mu}{\partial X}=\left\{2X-1-\frac{X(1-X)}{D}\left[2+4\frac{(1-X)}{S}(Y\,\mathrm{Cos}\,\psi)^2\right\}\Big/\left\{D\left(1-\frac{2X(1-X)}{D}\right)\right\}$$

$$\frac{1}{\mu}\frac{\partial\mu}{\partial Y}=\left\{X(1-X)Y\left[-2(1-\mathrm{Cos}^2\psi)+\frac{1}{S}(2(Y\,\mathrm{Sin}\,\psi)^2(1-\mathrm{Cos}^2\psi)+4(1-X)^2\mathrm{Cos}^2\psi\right]\right\}\Big/\left\{D^2\left(1-\frac{2X(1-X)}{D}\right)\right\}$$

$$\frac{1}{\mu}\frac{\partial\mu}{\partial\psi}=\left\{2X(1-X)Y^2\mathrm{Sin}\,\psi\,\mathrm{Cos}\,\psi\left[(Y\,\mathrm{Sin}\,\psi)^2-\frac{2}{S}(1-X)^2-1\right]\right\}\Big/\left\{D^2\left(1-\frac{2X(1-X)}{D}\right)\right\}$$

$$\frac{\partial\psi}{\partial\mathrm{Pr}}=\frac{\mathrm{Pr}\,\mathrm{Cos}\,\psi-\mu\mathrm{Nr}}{\mu^2\,\mathrm{Sin}\,\psi}$$

$$\frac{\partial\psi}{\partial P\theta}=\frac{P\theta\,\mathrm{Cos}\,\psi-\mu M\theta}{\mu^2\,\mathrm{Sin}\,\psi}$$

$$\frac{\partial\psi}{\partial P\theta}=\frac{P\phi\,\mathrm{Cos}\,\psi}{\mu^2\,\mathrm{Sin}\,\psi}$$

$$\frac{1}{\mu}\frac{\partial\mu}{\partial r}=\frac{1}{\mu}\frac{\partial\mu}{\partial X}*\frac{\partial N}{\partial r}*\left(\frac{8.98}{FKC}\right)^2+\frac{1}{\mu}\frac{\partial\mu}{\partial Y}*\left(-\frac{3Y}{R}\right)$$

$$\frac{1}{\mu}\frac{\partial\mu}{\partial\theta}=\frac{1}{\mu}\frac{\partial\mu}{\partial Y}*\left(-\frac{3}{2}M\theta\mathrm{Nr}Y\right)+\frac{1}{\mu}\frac{\partial\mu}{\partial\psi}\frac{\partial\psi}{\partial\theta}$$

$$\frac{1}{\mu}\frac{\partial\mu}{\partial\phi}=0$$

RETURN

Figure (IIA7.) Subroutine Partial MU
REF: (PP 3-24, 3-25 AFCRL 69-0123)

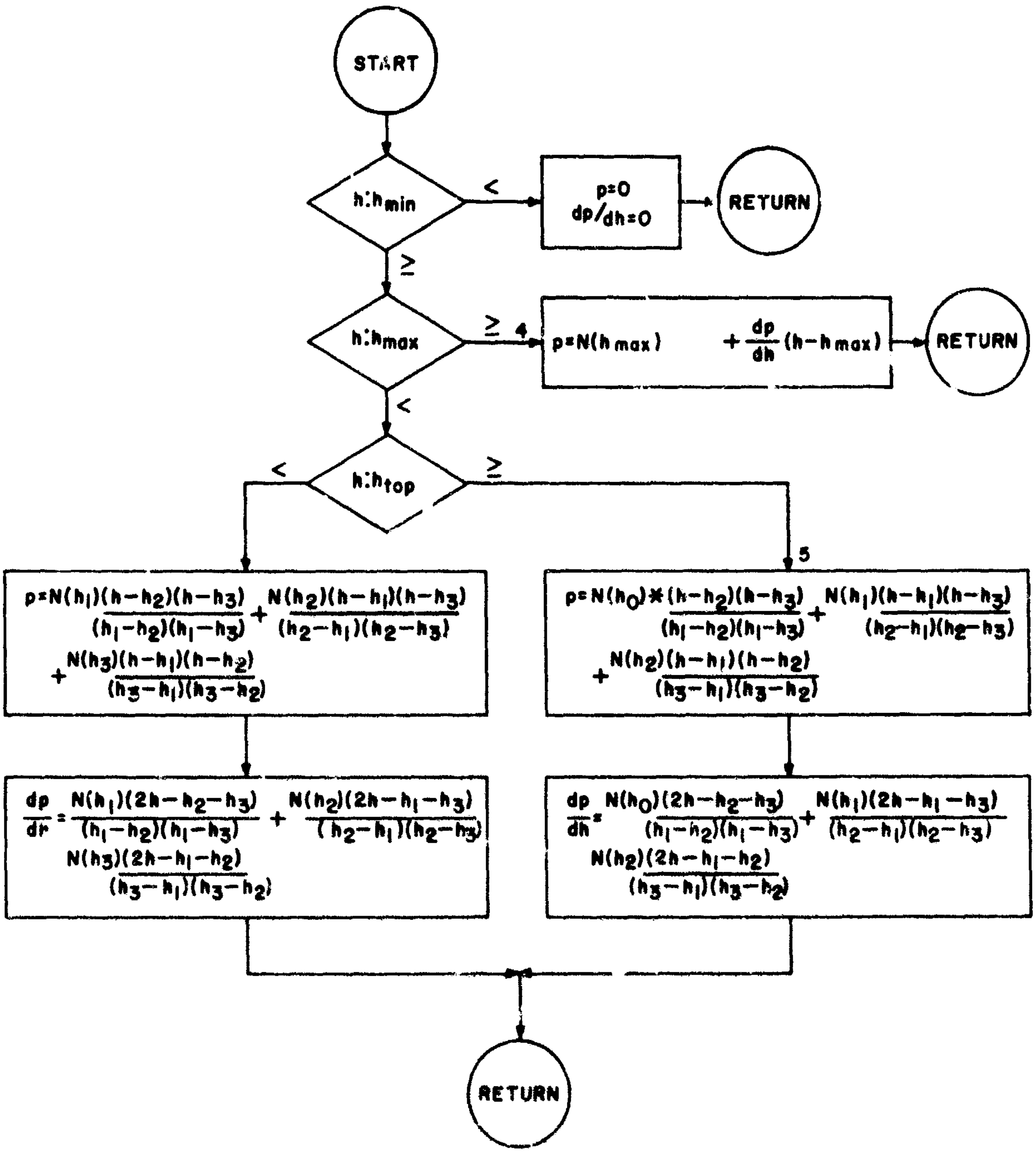


Figure (IIA8.) Subroutine Electron Density
REF. (Pg (3-22) AFCRL 69-0123)

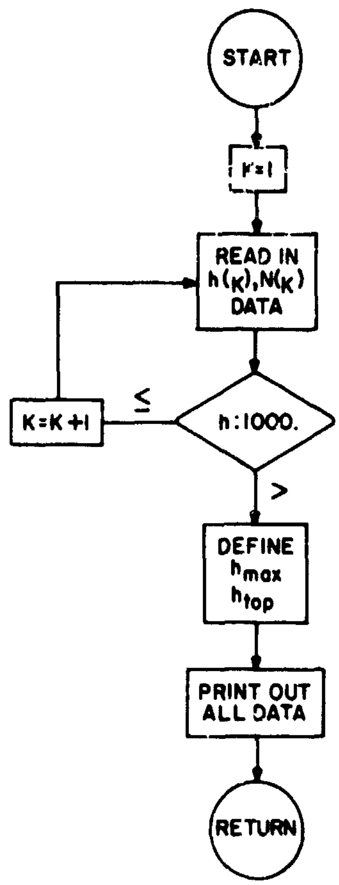


Figure (IIA9.) Subroutine Data Read-In

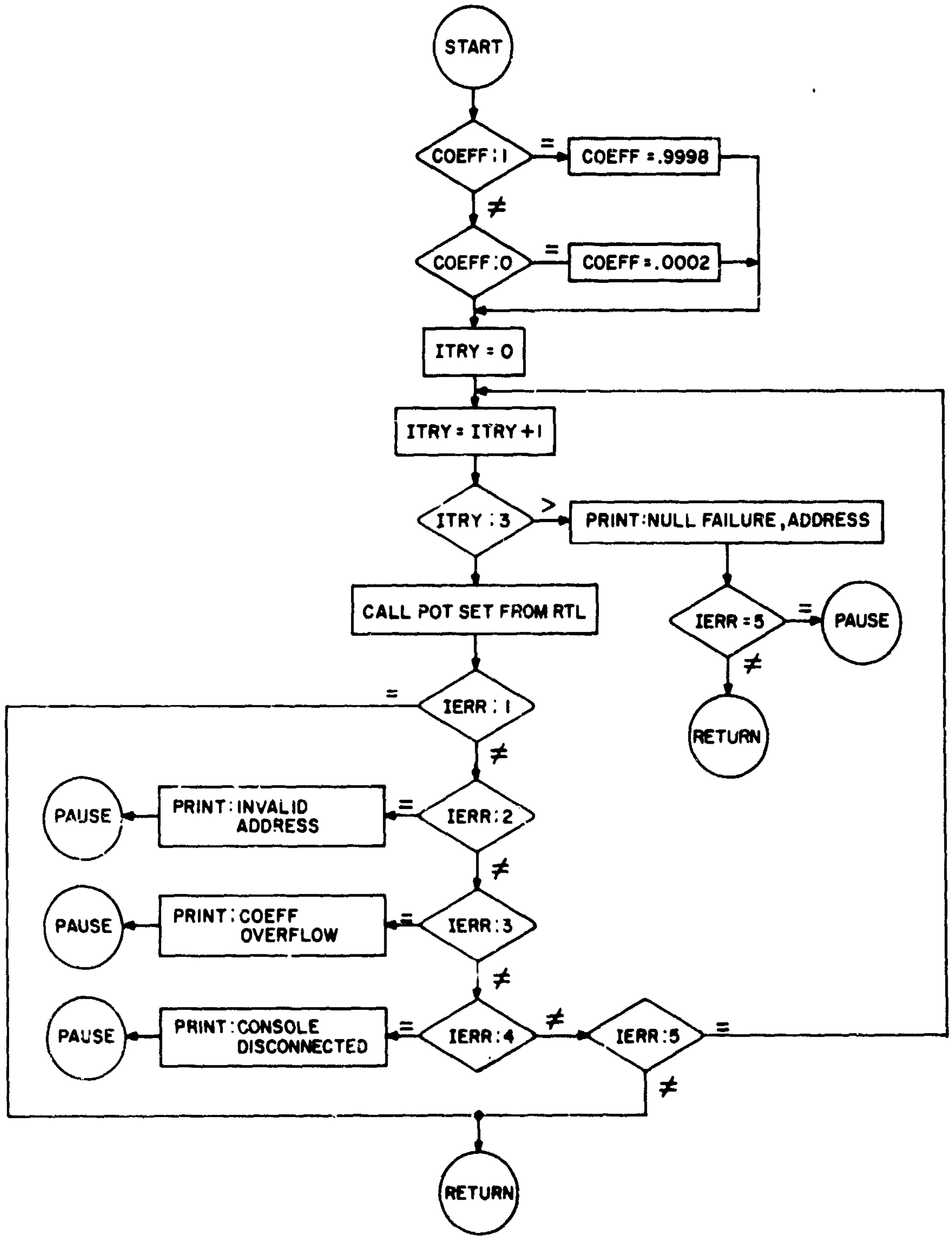


Figure (IIA10.) Subroutine Set Pot

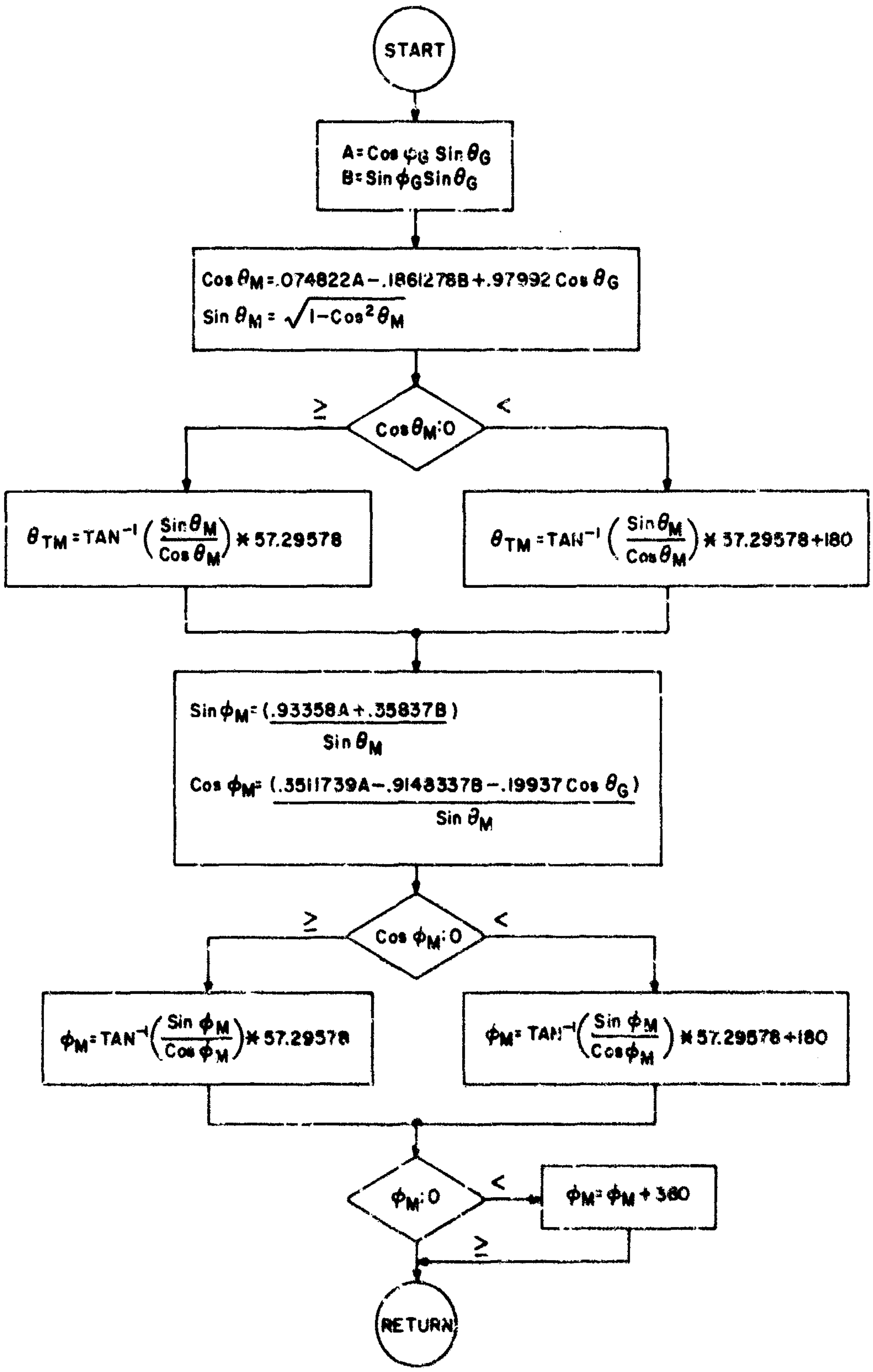


Figure (IIA1i.) Subroutine GEOTOM
REF: (P3-21, AFCRL 69-0123)

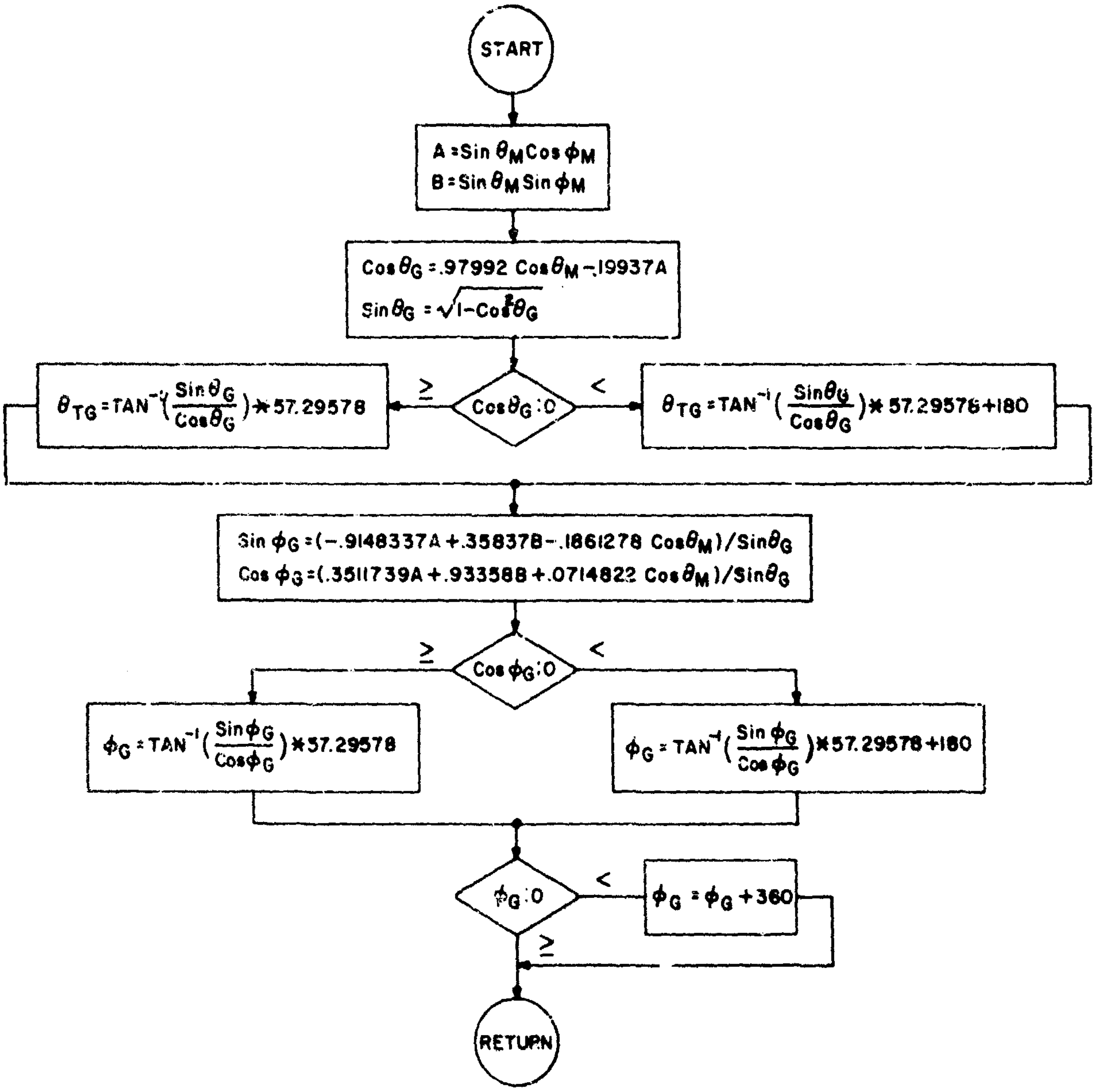


Figure (IIA12). Subroutine MAGTOG
REF: (PP 3-20, 3-21, AFCRL 69-0123)

IIA3.2 DIGITAL PROGRAM LISTING

```
$FO,.IU1,'4021
      DIMENSION ANGLIC(3),RINPUT(8),OUTPUT(13),SF(8)
      DIMENSION OUT1(100),OUT2(100),OUT3(100),OUT4(100)
      DIMENSION OUT5(100)
      DIMENSION AL(3),THF(6),HERR(6)
      COMMON/SCOOP/HGT(100),ED(100),KMAX,HMAX,HTOP
      COMMON/SCOOP1/SCAL(7),POTADR(3)
      COMMON/SCOOP2/H,THO,PHIO,ALPHA,BETA,IRUN
      COMMON/SCOOP3/A11,A12,A13,A21,A22,A23,A31,A32,A33
      COMMON/SCOOP4/PRO,PTHO,PPO
      COMMON/INFO/FKCT,FKC,SIGN
      COMMON/INFO1/R,COSTH,SINTH,COSPM,SINPM,THD,PHID,OPT,SOPT
      COMMON/INFO2/Y,YR,YTH,YSQ,NR,MTH,X,XR,OMX,OMX2,XOMX
      COMMON/INFO3/MUCK,PRN,PTHN,PPN,COSPS,COSP2,SINP2,SINPS
      COMMON/INFO4/YL,YT,YL2,YT2,MU,MUSQ,Z,Z2,ZR,ZMOD
      COMMON/INFO5/PSPR,PSPTH,PSPP,PSITH,MUPMU,MURMU,MUTMU,MUPHM,OPMUPO
      COMMON/INFO6/RM,M2,A0,A1,A2,A4,A5,A6,A7,A8,B0,B1,B2,B4,B5,B6,B7,B8
      COMMON/INFO7/D1,D2,A,B,RS4,RS,THS,S1,S2,DSQ,AA,AA2,BB,BB2
      COMMON/INFO8/RM4,THRM,M1,MUZMU,MUXMU,MUYMU
      COMMON/COORD/THTG,PHG,THTM,PHM
      EXTENDED POTADR
      REAL NR,MTH,MUCK,MU,MUSQ,MUPMU,MURMU,MUTMU,MUPHM,M2,MUZMU,MUXMU,
     1MUYMU,N
      LOGICAL INDICL,INDICM,INDICN,INDICK
      LOGICAL INTEG8
      LOGICAL INTEG,INTEG1,INTEG2,IDAVE
      LOGICAL IHIST,END,NOTBE
      EQUIVALENCE (SNPT1,H),(SNPT2,TH),(SNPT3,PHI)
      EQUIVALENCE (SNPT4,PR),(SNPT5,PTH),(SNPT6,PP)
      EQUIVALENCE (SNPT7,P)
      EQUIVALENCE (SNPT8,D)
      DATA AMPL/4HA000/
      DATA A11,A12,A13/ 0.3511739,-0.9148337,-0.19937/
      DATA A21,A22,A23/ 0.93358,0.35837,0.0/
      DATA A31,A32,A33/ 0.0714822,-0.1861278,0.97992/
      DATA (SF(J),J=1,8)/1000.0,3.4889,6.9778,2.0,2.0,2.0,5000.0,
     62000.0/
/       LDOB     ='40040,,12
C$$$$$$$$DATA READ IN AND VERIFICATION PRINT OUT OF ELECTRON DENSITY
      CALL DATA READ IN
C********READ SCALE FACTORS, POT ADDRESSES
      CALL ANALOG SETUP
C********READ IN RUN DATA AND PRINT OUT
 8400 CALL RUN DATA
C********RESET SENSE LINES 0,1,2,3
      DO 55 K=1,4
      J=K-1
      CALL QTEFFO(1,J,INDICL,IERR)
   55 CONTINUE
C********INITIALIZE STATE VARIABLES THRU COORDINATE XFORMATION
C**INITIALIZE TIME HISTORY ARRAYS
      ICOUNT=0.0
      END=.FALSE.
      NOTBE=.TRUE.
```

Preceding page blank

```
      TAU=0.0
      ITEST=1
      STEP=10.0
      ILAST=100
      OUT1(1)=0.0
      OUT2(1)=0.0
      OUT3(1)=0.0
      OUT4(1)=0.0
C**BEGIN ALPHA APPROX CALCULATION
      READ (5,610) THTARG,PHTARG,DTARG,THTRAN,PHTRAN,DALPH,DBETA,CONST
      WRITE (6,610) THTARG,PHTARG,DTARG,THTRAN,PHTRAN,DALPH,DBETA,CONST
  610 FORMAT(5X,2F10.4,F5.1,2F10.4,2F5.1,F10.3)
      BETA=221.43
      BORIG=BETA
      BETAE=270.0-BETA
      TOL1=0.01
      TOL2=10.0
      ZMOD=0.0
      I=0
  615 I=I+1
 7979 FORMAT(5H    I,5X,5HALPHA,5X,5H BETA,5X,5H  H  ,5X,5H THD ,5X,
     15H PHID//)
      IF(I.GT.3) GO TO 700
      ALPHA=12.5*FLOAT(I)
 7777 FORMAT(5X,F8.2)
      AL(I)=ALPHA
  650 THO=THTRAN
      PHIO=PHTRAN
      CALL INITIAL
      ITIME=0
      HZERO=0.0
C********CALCULATE AND SET IC POTS,H,THETAM,PHIM
      POT1=HZERO/1000.0
      POT2=THTM/200.0
      POT3=PHM/400.0
      CALL QPSO(1,IERR)
      CALL SETPOT (POTADR(1),POT1)
      CALL SETPOT (POTADR(2),POT2)
      CALL SETPOT(POTADR(3),POT3)
      CALL QRDALO(1,AMPL,DUMMY,IERR)
C********CALCULATE AND TRANSFER IC'S PR,PTH,PP
      PR=PRO
      PTH=PTHO
      PP=PPO
      TH=THO
      PHI=PHIO
      TMO=THO
      PMO=PHIO
      TM=TMO
      PM=PMO
      S=0.0
      TAU=0.0
C********CALCULATE INDEX OF REFRACTION
      CALL INDEX OF REFRACTION(H,TH,PHI,PR,PTH,PP)
```

```
      PRO=PRO*MU
      PTHO=PTHO*MU
      PPO=PPO*MU
C********SET UP BUFFER ARRAY FOR IC'S
      ANGLIC(1)=-PRO/2.0
      ANGLIC(2)=-PTHO/2.0
      ANGLIC(3)=-PPO/2.0
C********LOAD AND TRANSFER DAC'S
 2228 CALL QDALDO(13,3,ANGLIC,IERR,IECHL)
      IF(IERR.EQ.2) WRITE(6,81) IECHL
      IF(IERR.EQ.3) WRITE(6,82)
      CALL QDAXRO(12,4,IERR)
      IF(IERR.EQ.2) WRITE(6,83)
      CALL QICO(1,IERR)
      CALL QTEFFO(1,7,INTEG2,IEE)
      CALL QSYNCO(1,7,INTEG2)
      IPASS1=1
      GO TO 70
 1118 IPASS1=0
      WRITE(6,7725)ALPHA,BETAE
      TYPE 1108
      PAUSE
C*****TEST SLO FOR PRINTOUT
      CALL QTEFFO(1,0,IDAVE,IERR)
      IF(IDAVE) GO TO 70
 1936 IDAVE=.FALSE.
      TYPE 1109
 1108 FORMAT(42H BLIP FSW 1012 FOR IC PRINT...PRESS FLAG 8)
 1109 FORMAT(26H PRESS FLAG 8 TO CONTINUE.)
      PAUSE
/       SFL      ='63,,0
/       LDT      ='77777
      CALL QOPO(1,IERR)
      CALL QTEFFO(1,6,INTEG1,IEE)
      CALL QSYNCO(1,6,INTEG1)
C********ENTER HYBRID LOOP
   70 CONTINUE
/       STT      /ITIME
/       LDT      ='77777
/       SFL      ='62,,0
 6662 DELTAT=(32767-ITIME)*SCAL(7)
C********READ ADC'S
      CALL QSTREO(0,16,IERR)
      IF(IERR.EQ.2) WRITE (6,90)
      CALL QADCVO(0,8,RINPUT,IERR,IECHAN)
      IF(IERR.EQ.2) WRITE(6,91) IECHAN
      IF(IERR.EQ.3) WRITE(6,92)
      CALL QTRCKO (0,16,IERR)
      IF(IERR.EQ.2) WRITE (6,93)
C********ALL ANALOG INPUTS RECEIVED - UNSCALE ANALOG INPUTS
C***CHECK FOR H EQUAL 10. IF SO CHANGE SF(1)
      CALL QTEFFO(1,5,INTEG8,IEE)
      CALL QTEFFO(1,5,INTEG8,IERR)
      IF(INTEG8)SF(1)=10.0
```

```
C*****CHECK FO H LT -10 KM
      CALL QTEFFO(1,4,INTEG,IEE)
      IF(INTEG)SF(1)=1000.0
      SNPT1=SF(1)*RINPUT(1)
      SNPT2=SF(2)*RINPUT(2)
      SNPT3=SF(3)*RINPUT(3)
      SNPT4=SF(4)*RINPUT(4)
      SNPT5=SF(5)*RINPUT(5)
      SNPT6=SF(6)*RINPUT(6)
      SNPT7=SF(7)*RINPUT(7)
      SNPT8=SF(8)*RINPUT(8)
      SF(1)=1000.0
   50 CONTINUE
C********* TEST SL6 FOR OPERATE MODE OF ANALOG. IF ANALOG NOT IN THE
C********* OPERATE MODE SET DELTAT=0.0
      CALL QTEFFO(1,6,INTEG1,IEE)
      IF(.NOT.INTEG1) DELTAT=0.0
      CERC=SQRT(DTHC*DTHC+SINTH*SINTH*(DPC*DPC))
      TM=TM+DELTAT*DTHC
      PM=PM+DELTAT*DPC
      TH=TM
      PHI=PM
      S=S+6370.0*CERC*DELTAT
C********* TEST SL7 FOT IC MODE OF ANALOG, IF ANALOG IN IC REINITIALIZE
C********* THETA AND PHI
      CALL QTEFFO(1,7,INTEG2,IEE)
      IF(.NOT.INTEG2) GO TO 8765
      TM=TM0
      PM=PM0
      S=0.0
      TAU=0.0
 8765 CONTINUE
      IF(.NOT.END)GO TO 71
      TAU=TAU+DELTAT
      ICOUNT=TAU/STEP+0.001
      IF(ICOUNT.GT.ILAST)GO TO 71
      IF((ICOUNT+1).EQ.ITEST)GO TO 71
      ICP =ICOUNT+1
      OUT1(ICP)=TAU
      OUT2(ICP)=H
      OUT3(ICP)=S
      OUT4(ICP)=D
      OUT5(ICP)=P
      ITEST=ICP
   71 CONTINUE
C********CALCULATE MU(ELECTRON DENSITY(ELECTRON DENSITY DERIVATIVES,
C********MAGNETIC FIELD,AND MAGNETIC FIELD DERIVATIVES
      CALL INDEX OF REFRACTION(H,TH,PHI,PR,PTH,PP)
C********APPLICATION OF CONSTRAINT BETWEEN MU AND PR,PTH,PP
C********FOR DETERMINING CORRECT PR,PTH,PP
C********UNNORMALIZE PR
      PR=PRN*MU
C********TEST SL1 TO CHOOSE PP AS A FUNCTION OF PR,PTH,MU
      CALL QTEFFO(1,1,INDICL,KERRSS)
```

```
      IF (KERRSS.EQ.2) PRINT 4001
      IF (KERRSS.EQ.3) PRINT 4002
      IF(.NOT.INDICL) GO TO 969
C********HERE WE HAVE CHOSEN PP=F(PR,PTH,MU)
      PSIGN=-1.0
      IF (PP.GT.0.0) PSIGN=1.0
      RADCL=MUSQ*(1.-PTHN*PTHN)-PR*PR
      ARADCL=ABS(RADCL)
      PP=PSIGN* SQRT(ARADCL)
      GO TO 1969
C********TEST SL2 TO CHOOSE PTH AS A FUNCTION OF PR,PP,MU
  969 CALL QTEFFO(1,2,INDICM,ICRUS)
      IF(ICRUS.EQ.2) PRINT 4001
      IF(ICRUS.EQ.3) PRINT 4002
      IF(.NOT.INDICM) GO TO 1969
C********HERE WE HAVE CHOSEN PTH=F(PR,PP,MU)
      RPSIGN=-1.0
      IF(PTH.GT.0.0) RPSIGN=1.0
      RADCLI=MUSQ*(1.-PPN*PPN)-PR*PR
      ARADIC=ABS(RADCLI)
      PTH=RPSIGN*SQRT(ARADIC)
 1969 CONTINUE
C********PR,PTH,PP ARE NOW DETERMINED, WITH IR WITHOUT THE
C********APPLICATION OF THE CONSTRAINT
C********CALCULATION OF PARTIAL DERIVATIVES OF MU NOW FOLLOW
      CALL PARTIAL MU(PR,PTH,PP)
C********CALCULATION OF STATE VARIABLES FOLLOW
C********THETA DOT/C
      DTHC=(PTH/MUSQ-MUPMU*PSPTH)/R
C********PHI DOT/C
      DPC=(PP/MUSQ-MUPMU*PSPP)/(R*SINTH)
C********PR DOT/C
      DPRC=MURMU + PTH*DTHC + PP* DPC*SINTH
C********PTH DOT/C + PTH*R DOT/C
      PTHOTM=MUTMU/R + PP*COSTH*DPC
C********PPHI DOT/C + PPHI*R DOT/RC
      PPHDTM=(MUPHM/R - PP*COSTH*DTHC) / SINTH
C********LOAD DAC BUFFERS
      GROUP=1.+OPMUPO/MU
      OUTPUT(1)=SCAL(1)*50.0*DPRC
      OUTPUT(2)=SCAL(2)*500.0*PTHOTM
      OUTPUT(3)=SCAL(3)*500.0*PPHDTM
      OUTPUT(4)=SCAL(4)*DTHC*1000.0
      OUTPUT(5)=SCAL(5)*DPC*1000.0
      OUTPUT(6)=(PR/MUSQ)/2.0
      OUTPUT(7)=-SCAL(6)*MUPMU*PSPR/2.0
      OUTPUT(8)=5000.0*PTH/R
      OUTPUT(9)=5000.0*PP/R
      OUTPUT(10)=-MU/2.0
      OUTPUT(11)=DELTAT/100.0
      OUTPUT(12)=S/1000.0
      OUTPUT(13)=-GROUP/50.0
C********TEST SL0 FOR PRINT OUT CHECK
      CALL QTEFFO(1,0,INDICK,IERSS)
```

```
      IF(IERSS.EQ.2) PRINT 4001
      IF(IERSS.EQ.3) PRINT 4002
      IF(.NOT.(INDICK.OR.IDAVE)) GO TO 506
C*********IF INDICK TRUE (SLO SET) WE PRINT OUT
      N=X/FKCT
      DNR=XR/FKCT
C*********PRINTING FORMATS
      WRITE(6,77)
      WRITE (6,9140) PR,PTH,PP
      WRITE (6,9141) H,THD,PHID
      WRITE (6,9147) NR,MTH,COSPS
      WRITE(6,9145) Y,YR,X
      WRITE (6,9153) XR,D1,D2
      WRITE (6,9700) Z,ZR,Z2
      WRITE (6,9701) A,B,RS4
      WRITE (6,9702) RS,THS,S1
      WRITE (6,9703) S2,DSQ,AA
      WRITE (6,9704) AA2,BB,BB2
      WRITE (6,9705) RM4,RM,THRM
      WRITE (6,9706) M1,M2,MUSQ
      WRITE (6,9707) A0,A1,A2
      WRITE (6,9708) A4,A5,A6
      WRITE (6,9709)  A7,A8,B0
      WRITE (6,9710) B1,B2,B4
      WRITE (6,9711) B5,B6,B7
      WRITE (6,9712) B8,PSITH,MUXMU
      WRITE (6,9713) MUYMU,MUPMU,MUZMU
      WRITE(6,9159) PTHOTM,PPHDTM,GROUP
      WRITE (6,9156) PSPR,PSPTH,PSPP
      WRITE (6,9157) MURMU,MUTMU,MUPHM
      WRITE (6,9158) DTHC,DPC,DPRC
      WRITE (6,9154) MU,MUCK
      WRITE(6,8150) COSTH,SINTH
      WRITE(6,8151) OPT,SOPT
      WRITE(6,8152) N,DNR
      WRITE(6,9155) TH,PHI
      WRITE(6,77)
      WRITE(6,916)
  916 FORMAT(37H J          OUTPUT(J)             CHANNEL//)
      DO 816 J=1,13
      JCH=J-1
      WRITE (6,9160) J,OUTPUT(J),JCH
 9160 FORMAT(I3,6X,F14.6,11X,I2)
  816 CONTINUE
      WRITE(6,77)
      WRITE(6,9922)
 9922 FORMAT(37H J          RINPUT(J)             CHANNEL//)
      DO 47 J=1,6
      JCH=J-1
      WRITE(6,9160) J,RINPUT(J),JCH
   47 CONTINUE
      PAUSE
C*********DAC LOAD AND TRANSFER
  506 CALL QDALDO(0,13,OUTPUT,IERR4,IECHL)
```

```
      IF(IERR4.EQ.2) PRINT 81,IECHL
      IF(IERR4.EQ.3) PRINT 82
      CALL QDAXRO(0,16,IFRR5)
      IF(IERR5.EQ.2) PRINT 83
C********TEST SL3 FOR NEW RUN
      CALL QTEFFO(1,3,INDICN,IERR)
      IF(INDICN) GO TO 8400
      IF(IDAVE)GO TO 1936
      DIFF=ABS(PHTARG-PHID)
      IF(IPASS1.EQ.1)GO TO 1118
      IF(DIFF.GT.TOL1) GO TO 70
      CALL QHOLDO(1,IERR)
      IF(END)GO TO 913
      THE(I)=THD
      THORN=90.0-THD
      HERR(I)=10.0*H
      IF(I.LE.3) GO TO 7982
 7982 WRITE(6,7979)
      WRITE(6,7983)I,ALPHA,BETAE,H,THORN,PHID
 7983 FORMAT(I5,2X,F8.2,2X,F8.2,2X,F8.2,2X,F8.2,2X,F8.2/)
      DNEW=HERR(I)+ABS(CONST*(THE(I)-THTARG))
      DIFFD=ABS(DNEW)
      IF(DIFFD.LE.TOL2) GO TO 663
      IF(I-3) 615,690,700
  690 HORIG=20000.0
      DO 695 K=1,3
      IF(HERR(K).LT.HORIG) GO TO 692
      GO TO 695
  692 AORIG=AL(K)
      HORIG=HERR(K)
      THORIG=THE(K)
  695 CONTINUE
  651 I=4
      ALPHA=AORIG+DALPH
      BETA=BORIG
      GO TO 650
  700 IF(I-5)1701,1703,1703
 1701 DORIG=HORIG+ABS(CONST*(THORIG-THTARG))
      DADA=HERR(I)+CONST*(THE(I)-THTARG)
      PDPA=(DADA-DORIG)/DALPH
C***  TEST LOGICAL VARIABLE NOTBE TO DETERMINE BETA OMISSION
      IF(.NOT.NOTBE) GO TO 1754
      PDPBE=0.0
      GO TO 1753
 1754 I=5
      BETA=BORIG+DBETA
      ALPHA=AORIG
      GO TO 650
 1702 DBDB=HERR(I)+CONST*(THE(I)-THTARG)
      PDPBE=(DBDB-DORIG)/DBETA
 1753 GRADD2=PDPA*PDPA+PDPBE*PDPBE
      GRADD=SQRT(GRADD2)
      DALPH1=DALPH
      ADALF1=ABS(DALPH1)
```

```
      DALPH=-(DADA/GRADD2)*PDPA
      ADALF=ABS(DALPH)
      KAD=1
      IF(ADALF.GT.ADALF1) KAD=ADALF/ADALF1
      DALPH=DALPH/KAD
      DBETA=-(DADA/GRADD2)*PDPBE
C******** TEST NEW ALPHA,BETA
      ALPHA=ALPHA+DALPH
      BETA=BORIG+DBETA
      HORIG=HERR(I)
      I=I+1
      GO TO 650
C******** RE-INITIALIZE FOR CONTINUED SEARCH
 1703 AORIG=ALPHA
      BORIG=BETA
      THORIG=THD
      IF(END)GO TO 650
      IF(NOTBE) GO TO 1701
      GO TO 651
C******** STATEMENT 725 INDICATES SUCCESSFUL TARGETING
  663 TYPE 725
      END=.TRUE.
  725 FORMAT(29H THIS IS THE OPTIMUM RAY PATH)
 7725 FORMAT(7H ALPHA=,F8.2,10X,5HBETA=,F8.2/)
      PAUSE
      GO TO 1703
C**SET SLO FOR TIME HISTORY PRINTOUT
  913 TYPE 915
  915 FORMAT(25H SET SLO FOR TIME HISTORY)
      PAUSE
C***TIME HISTORY ARRAY CHECK FOR PRINTOUT
      CALL QTEFFO(1,0,IHIST,IERR)
      IF(IHIST)GO TO 914
  926 END=.FALSE.
      CALL QICO(1,IERR)
      GO TO 70
C****PRINTOUT OF TIME HISTORY
  914 WRITE(6,2121)
 2121 FORMAT(10X,15H H=ALTITUDE(KM)/,10X,28H S=GREAT CIRCLE DISTANCE(KM)
     6/,10X,17H D=PHASE PATH(KM)/,10X,17H P=GROUP PATH(KM)/)
      DO 89 ICP=1,ITEST
      WRITE(6,80)ICP,OUT1(ICP),OUT2(ICP)
      WRITE(6,84, OUT3(ICP),OUT4(ICP),OUT5(ICP)
   80 FORMAT(5H ICP=I2,5H TAU=F10.2,10X,3H H=F10.6)
   84 FORMAT(8X,3H S=F10.6,10X,3H D=F10.6,10X,3H P=F10.6)
   89 CONTINUE
      TYPE 444
  444 FORMAT(32H TIME HISTORY PRINTED, RESET SLO)
      PAUSE
      GO TO 926
C********FORMAT STATEMENTS
 90   FORMAT (31H BLK ADR ERROR IN STORE ROUTINE)
   91 FORMAT (23H ADC CHANNEL OVERLOAD =,I5)
   92 FORMAT (21H NOW-EXISTING CHANNEL)
```

```
   93 FORMAT (31H BLK ADR ERROR IN TRACK ROUTINE)
   81 FORMAT (25H OVERFLOW IN DAC CHANNEL ,I5)
   82 FORMAT (25H NON-EXISTING DAC CHANNEL)
   83 FORMAT (22H DAXFR BLOCK ADR ERROR)
 4001 FORMAT(21H CONSOLE DISCONNECTED)
 4002 FORMAT(33H NON-EXISTING COMPONENT REQUESTED)
   77 FORMAT(1H1)
 8150 FORMAT(7H COSTH=,F7.4,11X,7H SINTH=,F7.4)
 8151 FORMAT(5H OPT=,F8.4,12X,6H SOPT=,F8.4)
 8152 FORMAT(3H N=,E13.6,9X,5H DNR=,E13.6)
 9155 FORMAT(4H TH=,F10.6,11X,5H PHI=,F10.6)
 9147 FORMAT(4H NR=,F7.5,14X,5H MTH=,F7.5,13X,7H COSPS=,F7.5)
 9140 FORMAT(4H PR=,F7.5,14X,5H PTH=,F7.5,13X,4H PP=,F7.5)
 9141 FORMAT(3H H=,F12.6,10X,5H THD=,F12.6,8X,6H PHID=,F12.6)
 9145 FORMAT(3H Y=,F7.5,15X,4H YR=,F9.5,12X,3H X=,F10.5)
 9153 FORMAT(4H XR=,E13.6,8X,4H D1=,E13.6,8X,4H D2=,E13.6)
 9154 FORMAT(4H MU=,F7.5,14X,6H MUCK=,F7.5)
 9156 FORMAT(6H PSPR=,F12.6,7X,7H PSPTH=,F12.6,6X,6H PSPP=,F12.6)
 9157 FORMAT (7H MURMU=,F12.6.6X,7H MUTMU=,F12.6,6X,7H MUPHM=,F12.6)
 9158 FORMAT(6H DTHC=,F12.6,7X,5H DPC=,F12.6,8X,6H DPRC=,F12.6)
 9159 FORMAT(8H PTHOTM=,F12.6,5X,8H PPHDTM=,F12.6,5X,7H GROUP=,F7.4)
 9700 FORMAT(3H Z=,E13.6,9X,4H ZR=,E13.6,8X,4H Z2=,E13.6)
 9701 FORMAT(3H A=,E13.6,9X,3H B=,E13.6,9X,5H RS4=,E13.6)
 9702 FORMAT 4H RS=,E13.6,8X,5H THS=,F13.6,7X,4H S1=,E13.6)
 9703 FORMAT(4H S2=,E13.6,8X,5H DSQ=,E13.6,7X,4H AA=,E13.6)
 9704 FORMAT(5H AA2=,E13.6,7X,4H BB=,E13.6,8X,5H BB2=,E13.6)
 9705 FORMAT(5H RM4=,E13.6,7X,4H RM=,E13.6,8X,6H THRM=,E13.6)
 9706 FORMAT(4H M1=,E13.6,8X,4H M2=,E13.6,8X,6H MUSQ=,E13.6)
 9707 FORMAT(4H A0=,E13.6,8X,4H A1=,F13.6,8X,4H A2=,E13.6)
 9708 FORMAT(4H A4=,E13.6,8X,4H A5=,E13.6,8X,4H A6=,E13.6)
 9709 FORMAT(4H A7=,E13.6,3X,4H A8=,E13.6,8X,4H B0=,E13.6)
 9710 FORMAT(4H B1=,E13.6,8X,4H B2=,E13.6,8X,4H B4=,E13.6)
 9711 FORMAT(4H B5=,E13.6,8X,4H B6=,E13.6,8X,4H B7=,E13,6)
 9712 FORMAT(4H B8=,E13.6,8X,7H PSITH=,E13.6,5X,7H MUXMU=,E13.6)
 9713 FORMAT(7H MUYMU=,E13.6,5X,7H MUPMU=,E13.6,5X,7H MUZMU=,E13.6)
      END
```

```
 9158 FORMAT(6H DTHC=,F12.6,7X,5H DPC=,F12.6,8X,6H DPRC=,F12.6)
 9159 FORMAT(9H PTHOTM=,F12.6,5X,8H PPHOTM=,F12.6,5X,7H GROUP=,F7.4)
      END
.
.

      SUBROUTINE ANALOG SETUP
      COMMON/SCOOP1/SCAL(7),POTADR(3)
      EXTENDED POTADR
      READ(5,77) (SCAL(K),K=1,7)
      WRITE(6,77)(SCAL(K),K=1,7)
   77 FORMAT(7F11.5)
      READ(5,777) (POTADR(J),J=1,3)
  777 FORMAT (A4,4X,A4,4X,A4)
      RETURN
      END
.
.

      SUBROUTINE RUN DATA
      COMMON/INFO/FKCT,FKC,SIGN
      COMMON/SCOOP2/H,THO,PHIO,ALPHA,BETA,IRUN
 8400 READ(5,34) IRUN
   34 FORMAT(I8)
      READ(5,7033) H,THO,PHIO,ALPHA,BETA
 7033 FORMAT(5F15.7)
      READ(5,7033) SIGN,FKC
      WRITE(6,7000)
 7000 FORMAT(1H1)
      WRITE(6,7010) IRUN
 7010 FORMAT(12H RUN NUMBER ,I3//)
      WRITE(6,7011) SIGN,FKC
 7011 FORMAT(6H SIGN=,F4.1,6X,7HFREQKC=,F8.2/)
      WRITE(6,7012) H,THO,PHIO
 70120FORMAT(3H H=,F7.2,6X,7HTHETAG=,F6.2,3X,5HPHIG=,
     1F6.2/)
      WRITE(6,7013) ALPHA,BETA
 7013 FORMAT(7H ALPHA=,F6.2,3X,5HBETA=,F6.2/)
      RETURN
      END
.
.

      SUBROUTINE INITIAL
      COMMON/INFO/FKCT,FKC,SIGN
      COMMON/SCOOP2/H,THO,PHIO,ALPHA,BETA,IRUN
      COMMON/SCOOP3/A11,A12,A13,A21,A22,A23,A31,A32,A33
      COMMON/SCOOP4/PRO,PTHO,PPO
      COMMON/COORD/THTG,PHG,THTM,PHM
      FKCSQ=FKC*FKC
      FKCT= 80.6451/FKCSQ
      SINTHG=SIN(THO/57.29578)
      COSTHG=COS(THO/57.29578)
      SINPHG=SIN(PHIO/57.29578)
      COSPHG=COS(PHIO/57.29578)
      CALL GEOTOM(SINTHG,COSTHG,SINPHG,COSPHG)
      THO=THTM/57.29578
      PHIO=PHM/57.29578
C********THO(PHIO ARE NOW GEOMAGNETIC COORDINATES
      SINPM=SIN(PHIO)
      COSPM=COS(PHIO)
      SINTM=SIN(THO)
      COSTM=COS(THO)
      C1=A11*COSTM*COSPM+A21*COSTM*SINPM-A31*SINTM
      C2=A12*COSTM*COSPM+A22*COSTM*SINPM-A32*SINTM
      C3=A13*COSTM*COSPM+A23*COSTM*SINPM-A33*SINTM
```

```
      D1=-A11*SINPM+A21*COSPM
      D2=-A12*SINPM+A22*COSPM
      D3=-A13*SINPM+A23*COSPM
      THGTHM=COSTHG*COSPHG*C1+COSTHG*SINPHG*C2-SINTHG*C3
      PHGTHM=-SINPHG*C1+COSPHG*C2
      THGPHM=COSTHG*COSPHG*D1+COSTHG*SINPHG*D2-SINTHG*D3
      PHGPHM=-SINPHG*D1+COSPHG*D2
      PR0=SIN(ALPHA/57.29578)
      PTHG0=COS(ALPHA/57.29578)*SIN(BETA/57.29578)
      PPG0=COS(ALPHA/57.29578)*COS(BETA/57.29578)
      PTH0=THGTHM*PTHG0+PHGTHM*PPG0
      PP0=THGPHM*PTHG0+PHGPHM*PPG0
      RETURN
      END


      SUBROUTINE INDEX OF REFRACTION (H,TH,PHI,PR,PTH,PP)
      COMMON/COORD/THTG,PHG,THTM,PHM
      COMMON/INFO/FKCT,FKC,SIGN
      COMMON/INFO1/R,COSTH,SINTH,COSPM,SINPM,THD,PHID,OPT,SOPT
      COMMON/INFO2/Y,YR,YTH,YSQ,NR,MTH,X,XR,OMX,OMX2,XCMX
      COMMON/INFO3/MUCK,PRN,PTHN,PPN,COSPS,COSP2,SINP2,SINPS
      COMMON/INFO4/YL,YT,YL2,YT2,S,C,MUSQ,MU
      REAL NR,MTH,MUCK,MUSQ,MU,N
      R=H+6371.0
      COSTH=COS(TH)
      SINTH=SIN(TH)
      COSPM=COS(PHI)
      SINPM=SIN(PHI)
      CALL MAGTOG(SINTH,COSTH,SINPM,COSPM)
C********THD, PHID ARE GEOGRAPHIC COORDINATES
      THD=THTG
      PHID=PHG
      OPT=1. + 3.* COSTH**2
      SOPT=SQRT(OPT)
      Y=880.*(SOPT*(6370./R)**3)/FKC
C********Y IS THE NORMALIZED MAGNETIC FIELD
      YR= - 3.0* Y/R
      YTH=-3.*COSTH*SINTH*Y/OPT
C********YR,YTH ARE DERIVATIVES WRT R,TH, RESPECTIVELY
      YSQ = Y*Y
      NR= 2.*COSTH/SOPT
C********Y*NR IS THE MAGNETIC FIELD COMPONENT IN THE R DIRECTION
      MTH=SINTH/SOPT
C********Y*MTH IS THE MAGNETIC FIELD COMPONENT IN THE TH DIRECTION
C********WE NOW CALCULATE ELECTRON DENSITY AND ITS DERIVATIVES
      CALL ELECTRON DENSITY (H,N,DNR)
   40 X=N*FKCT
      XR=DNR*FKCT
      OMX=1.-X
   30 OMX2=OMX*OMX
      XOMX=X*OMX
      MUCK=SQRT(PR*PR+PTH*PTH+PP*PP)
C********NORMALIZE PR,PTH,PP
      PRN=PR/MUCK
      PTHN=PTH/MUCK
      PPN=PP/MUCK
      COSPS=PRN*NR+PTHN*MTH
      COSP2=COSPS*COSPS
      SINP2=1.-COSP2
      SINPS=SQRT(SINP2)
      YL=Y*COSPS
      YT=Y*SINPS
      YL2=YL*YL
      YT2=YT*YT
```

```
      S=SIGN*SQRT(YT2*YT2+4.0*OMX2*YL2)
      D=2.*OMX+S-YT2
      MUSQ=1.0-2.0*XOMX/D
  300 MU=SQRT(MUSQ)
      RETURN
      END
*
*
      SUBROUTINE PARTIAL MU (PR,PTH,PP)
      COMMON/INFO1/R,COSTH,SINTH,COSPM,SINPM,THD,PHID,OPT,SOPT
      COMMON/INFO2/Y,YR,YTH,YSQ,NR,MTH,X,XR,OMX,OMX2,XOMX
      COMMON/INFO3/MUCK,PRN,PTHN,PPN,COSPS,COSP2,SINP2,SINPS
      COMMON/INFO4/YL,YT,YL2,YT2,S,D,MUSQ,MU
      COMMON/INFO5/PSPR,PSPTH,PSPP,MUPMU,MURMU,MUTMU,MUPHM
      REAL NR,MTH,MUCK,MUSQ,MU
      REAL M2YSP,MUXMU,MUYMU,MUPMU,MURMU,MUTMU,MUPHM
      PSITH=2.*(PR*MTH-PTH*NR)/(MU*SINPS*OPT)
      DSQ=D*D
      M2YSP=MUSQ*SINPS
      RMUXM1=2.0+4.0*OMX*YL2/S
      RMUXM2=XOMX*RMUXM1/D
      MUXMU=(2.0*X-1.0-RMUXM2)/(D*MUSQ)
      RMUYM1=2.0*YT2*SINP2+4.0*OMX2*COSP2
      RMUYM2=-2.0*SINP2+RMUYM1/S
      MUYMU=XOMX*Y*RMUYM2/(DSQ*MUSQ)
      RMUPM1=(YT2-2.0*OMX2)/S-1.0
      MUPMU=XOMX*2.0*YL*YT*RMUPM1/(MUSQ*DSQ)
C     REAPARTIALS OF PSI WRT PR,PTH,PP
      PSPR=(PR*COSPS-MU*NR)/M2YSP
      PSPTH=(PTH*COSPS-MU*MTH)/M2YSP
      PSPP=PP*COSPS/M2YSP
C     REASPATIAL DERIVATIVES OF MU
      MURMU=MUXMU*XR+MUYMU*YR
      MUTMU=MUYMU*YTH+MUPMU*PSITH
      MUPHM=0.0
      RETURN
      END
*
*
      SUBROUTINE ELECTRON DENSITY (H,N,DNR)
      COMMON/SCOOP/HGT(100),ED(100),KMAX,HMAX,HTOP
      REAL N
      HMIN=HGT(1)
      IF(H.GE.HMIN) GO TO 1
      N=0.0
      DNR=0.0
      RETURN
C********** INTERPOLATION BY PARABOLA
    1 IF(H.GE.HMAX) GO TO 4
      IF(H.GE.HTOP) GO TO 5
C********** INTERVAL SEARCH
      DO 10 J=1,KMAX
      XMXI=H-HGT(J)
      XIPMX=HGT(J+1)-H
      IF((XMXI.GE.0.0).AND.(XIPMX.GE.0.0)) GO TO 3
   10 CONTINUE
    3 I=J
      T1=HGT(I)
      T2=HGT(I+1)
      T3=HGT(I+2)
      T12T13=(T1-T2)*(T1-T3)
      T21T23=(T2-T1)*(T2-T3)
      T31T32=(T3-T1)*(T3-T2)
C********** BASE VECTORS FOLLOW P1(P2(P3
```

```
      P1=(H-T2)*(H-T3)/T12T13
      P2=(H-T1)*(H-T3)/T21T23
      P3=(H-T1)*(H-T2)/T31T32
C********* ELECTRON DENSITY
      N=ED(I)*P1+ED(I+1)*P2+ED(I+2)*P3
      N=1000.0*N
C********** ELECTRON DENSITY DERIVATIVE
      A1=(2.*H-T2-T3)/T12T13
      A2=(2.*H-T1-T3)/T21T23
      A3=(2.*H-T1-T2)/T31T32
      DNR=A1*ED(I)+A2*ED(I+1)+A3*ED(I+2)
      DNR=1000.0*DNR
      RETURN
    5 T1=HGT(I-1)
      T2=HGT(I)
      T3=HGT(I+1)
      T12T13=(T1-T2)*(T1-T3)
      T21T23=(T2-T1)*(T2-T3)
      T31T32=(T3-T1)*(T3-T2)
      P1=(H-T2)*(H-T3)/T12T13
      P2=(H-T1)*(H-T3)/T21T23
      P3=(H-T1)*(H-T2)/T31T32
C********* ELECTRON DENSITY
      N=ED(I-1)*P1+ED(I)*P2+ED(I+1)*P3
      N=1000.0*N
C********** ELECTRON DENSITY DERIVATIVE
      A1=(2.*H-T2-T3)/T12T13
      A2=(2.*H-T1-T3)/T21T23
      A3=(2.*H-T1-T2)/T31T32
      DNR=A1*ED(I-1)+A2*ED(I)+A3*ED(I+1)
      DNR=1000.0*DNR
      RETURN
C********** H GT HMAX
    4 N=ED(KMAX)*1000.0+DNR*(H-HMAX)
      RETURN
      END
C
C
      SUBROUTINE DATA READ IN
      COMMON/SCOOP/HGT(100),ED(100),KMAX,HMAX,HTOP
      DO 10 K=1,100
      READ(5,1) HGT(K),ED(K)
    1 FORMAT(F8.3,10X,F10.4)
C******** LAST DATA CARD HAS H=2000
      IF(HGT(K).GT.1999.0) GO TO 11
   10 CONTINUE
C********** KMAX IS THE TOTAL NUMBER OF PAIRED DATA POINTS
   11 KMAX=K-1
      HMAX=HGT(KMAX)
      HTOP=HGT(KMAX-1)
C********** VERIFY DATA BY PRINT OUT
      WRITE(6,2)
    2 FORMAT(1H1)
      WRITE(6,3)
    3 FORMAT(2H  ,4X,5HH(KM),10X,12HELECTRONS/CC//)
      DO 20 K=1,KMAX
      WRITE(6,4) HGT(K),ED(K)
   20 CONTINUE
    4 FORMAT(F11.3,10X,E13.5)
      RETURN
      END
C
C
      SUBROUTINE GEOTOM(SINTHG,COSTHG,SINPHG,COSPHG)
```

```
      COMMON/COORD/THTG,PHG,THTM,PHM
      A=COSPHG*SINTHG
      B=SINPHG*SINTHG
      COSTM=0.0714822*A-0.1861278*B+0.97992*COSTHG
      SINTM=SQRT(1.0-COSTM*COSTM)
      IF(COSTM.LT.0.0) GO TO 5
      THTM=ATAN(SINTM/COSTM)*57.29578
      GO TO 4
    5 THTM=ATAN(SINTM/COSTM)*57.29578+180.0
    4 SINPM=(0.93358*A+0.35837*B)/SINTM
      COSPM=(0.3511739*A-0.9148337*B-0.19937*COSTHG)/SINTM
      IF(COSPM.LT.0.0) GO TO 6
      PHM=ATAN(SINPM/COSPM)*57.29578
      GO TO 7
    6 PHM=ATAN(SINPM/COSPM)*57.29578+180.0
    7 IF(PHM.LT.0.0) PHM=PHM+360.0
      RETURN
      END
EOF


      SUBROUTINE SETPOT(ADDR,COEF)
      EXTENDED ADDR
      IF(COEF.EQ.1.0000) COEF=0.9998
      IF(COEF.EQ.0.0000) COEF=0.0002
      ITRY=0
    2 ITRY=ITRY+1
      IF(ITRY.GT.3)GOTO4
      CALL OSTPTO(1,ADDR,COEF,IERR)
      IF(IERR.EQ.1) RETURN
      IF(IERR.EQ.2) WRITE(6,20) ADDR
      IF (IERR.EQ.2) PAUSE
      IF (IERR.EQ.3) WRITE(6,30) ADDR
      IF(IERR.EQ.3) PAUSE
      IF (IERR.EQ.4) WRITE(6,40) ADDR
      IF (IERR.EQ.4) PAUSE
      IF(IERR.EQ.5)GOTO2
      RETURN
    4 WRITE(6,50)ADDR
      IF (IERR.EQ.5) PAUSE
   20 FORMAT(17H INVALID ADDRESS ,A4)
   30 FORMAT(16H COEFF OVERFLOW ,A4)
   40 FORMAT (21H CONSOLE DISCONNECTED)
   50 FORMAT (14H NULL FAILURE ,A4)
      RETURN
      END


      SUBROUTINE MAGTOG(SINTHM,COSTHM,SINPHM,COSPHM)
      COMMON/COORD/THTG,PHG,THTM,PHM
      A=SINTHM*COSPHM
      B=SINTHM*SINPHM
      COSTG=0.97992*COSTHM-0.19937*A
      SINTG=SQRT(1.0-COSTG*COSTG)
      IF(COSTG.LT.0.0) GO TO 15
      THTG=ATAN(SINTG/COSTG)*57.29578
      GO TO 14
   15 THTG=ATAN(SINTG/COSTG)*57.29578+180.0
   14 SINPG=(-0.9148337*A+0.35837*B-0.1861278*COSTHM)/SINTG
      COSPG=(0.3511739*A+0.93358*B+0.0714822*COSTHM)/SINTG
      IF(COSPG.LT.0.0) GO TO 16
      PHG=ATAN(SINPG/COSPG)*57.29578
      GO TO 17
   16 PHG=ATAN(SINPG/COSPG)*57.29578+180.0
   17 IF(PHG.LT.0.0) PHG=PHG+360.0
      RETURN
```

IIB1. PROCEDURE FOR RUNNING THE RAY TRACING PROGRAM (I)

i) Load main program plus subroutines and data into the hopper.

ii) Type $GO on the console typewriter.

Sequence of events

a) Program plus data will be read in.

b) After reading data, the electron density profile used will be printed plus the target and transmitter parameters.

c) Next the run number will be printed with the initial conditions which specify an ordinary or an extraordinary ray.

d) This should be followed by the IC mode of the analog computer and the console typewriter output message: "BLIP FSW 1012 FOR IC PRINT... PRESS FLAG 8". This will be followed by a PAUSE.

iii) Release the PAUSE. There are two courses of action which could follow:

a) If FSW 1012 was "blipped" an IC print will follow ending with a PAUSE. Releasing this PAUSE will cause the typewriter to type, "PRESS FLAG 8 TO CONTINUE".

b) If FSW 1012 was not "blipped" the typewriter will immediately type, "PRESS FLAG 8 TO CONTINUE" after releasing the PAUSE of step ii)d.

c) Either (a) or (b) of this section is followed by a PAUSE.

iv) Again release the PAUSE.

Sequence of events

a) Once the tolerance on PHI is satisfied, i.e. the end of one run, the analog computer will be placed into HOLD.

b) The line printer will write the headers for "I, ALPHA, BETA, H, THD, PHID" followed by their values at the end of the run.

c) Next the IC potentiometers will be set (H, PHID, THD), the analog computer will go to IC and the typewriter message of step ii)d will be repeated. Also, as a verification of the next β, α pair their modified values will be printed on the line printer.

v) From this point steps ii)d through iv) are repeated until convergence criteria are met.

vi) When the ray is the optimum one within the constraints, the typewriter output is "THIS IS THE OPTIMUM RAY PATH". A PAUSE will follow.

vii) Release the PAUSE and repeat steps ii)d through iv) for this ray. (We are repeating the optimum ray for time history storage.)

a) At the end of this ray path the typewriter message is, "SET SLO FOR TIME HOSTORY PRINT" followed by a PAUSE.*

viii) Set FSW 1012 and release the PAUSE.

a) Time history printout will follow.

b) Typewriter message at conclusion of print is, "TIME HISTORY PRINTED RESET SLO". (The last message is to ensure that the switch is not left in the set position for the next run.)

ix) This is conclusion of one run for a given electron density profile.

x) To load new electron density data, replace the old electron density profile with a new one in the data deck. The end of electron density data signal (H = 2000) must be retained.

*NOTE: Since line zero (SLO) is FSW 1012.

IIB2. CHANGES TO ANALOG BOARD REQUIRED TO CONVERT FROM AUTOMATIC TO MANUAL RAY TRACING

Amp 210 goes directly into trunk line 330 for the manual case. In the automatic case amp 414 goes into trunk 330 and amp 210's output into trunk 330 is removed.

For the automatic case, the output of amp Ø14 should input comparator ØØØ.

For the manual case, the output of amp 812 should input comparator ØØØ.

For both cases the timer in the 8400 should be set at 10 μsec.

Other than the above there is no change to the analog board. The static test program is applicable to both cases. Essentially all that is done by the above is to remove the H optimum circuit. The pots will be set by the static test program but it doesn't make any difference since amp 210 bypasses the optimum circuit.

IIB3. MANUAL RAY TRACE

PHASE II

SWITCH AND SENSE LINE

ASSIGNMENT

Sense Line	F. Switch	Function
0	1012	IC Printout Check
*1	1211(1)	$P\phi = f(Pr, P\theta, \mu)$
*2	1053(0)	$P\theta = f(Pr, P\phi, \mu)$
3	1013	New Run
4	Not Used	
5	Not Used	
*6	OP	Hold Simulated
*7	IC	IC Simulated
--	411	L-Alt. C-Phase Path
	(For X-Y Plotter)	R-Group Path

*Asterisks denote hard wired sense lines.

IIB4. PROCEDURE II

(ORIGINAL MANUAL RAY TRACING)

i) Place deck (Main plus subroutines) in the hopper

ii) $GO on console typewriter

<u>Sequence of Events</u>

a) Program plus data will be read in

b) After reading data (see data deck org)

1) The electron density profile will be output on the line printer

2) The IC pots (C310, C701, C910) will be set

3) The analog computer will go to the I.C. mode

iii) To make sure the program has been loaded properly, blip function switch 1012. This will give you an IC printout followed by a Fortran pause (Flag 8 high). Release the pause, more printout will insue with a second pause. Release the second pause.

iv) Go to the analog console and manually place the analog computer in the operate mode. The ray will run and the analog computer will hold at the distance 607 KM. i.e. is distance from Greenbelt, MD. to AFCRL, Bedford, Mass. If at this point a printout is desired follow the steps in iii After this place the analog computer into IC. This is the end of one ray.

v) To reinitialize the program for another ray (i.e. increase alpha) Flip function switch 1013, to the left. The new data cards will be read, their values printed, the pots set and the analog will go to IC. From here repeat step iv).

IIB5. DATA DECK ORGANIZATION

Data is read into the computer at the beginning of each run. Initial conditions for each run are contained on three cards as indicated below:

CARD (1): IRUN (I8)
IRUN is the run number. It is fixed point and can be up to eight digits in length.

CARD (2): H, THO, PHIO, ALPHA, BETA (5F15.7)
The initial value of Height, Theta, Phi, Alpha and Beta are on card 2 in floating point form. They can be defined with up to seven decimal places.

CARD (3): SIGN, FKC (2F15.7)
SIGN indicates the type of ray. For an ORDINARY ray, SIGN = 1.0

For an EXTRAORDINARY ray, SIGN = -1.0.
FKC is the frequency of the ray in kilohertz.

III. AN OPTICALLY SCALED NUCLEAR EMULSIVE TRACK TRACER

Preceding page blank

III.1.0 INTRODUCTION

A nuclear emulsion is a material which records, photographically, the tracks of charged particles. An ionizing particle, one with sufficient energy, on encountering a crystal of nuclear emulsion renders it developable. After development and further processing, the paths of charged particles that penetrated the emulsion are visible through a microscope as trails of minute grains. This trail of grains represents a three-dimensional image of the charged particle's path.

The major application of nuclear emulsion is in experimental physics. Emulsions provide the means with which interactions between charged particles may be observed.

The instrument used for analyzing the behavior of tracks in emulsions is the microscope. The path of a high energy particle through an emulsion is presently scanned by human scanners using microscopes. This scanning process, as performed by humans is tedious and subject to errors caused by fatigue of the human scanner.

This report describes the concepts developed for a track tracing system and their embodiement within an optically scaled breadboard model for automating the scanning and analysis of nuclear emulsions. The results obtained with this breadboard model established the credence of these concepts.

III.2.0 REQUIREMENTS OF AN AUTOMATED SCANNING SYSTEM

Any system for automating the scanning process for nuclear emulsions must coordinate and integrate the following tasks:

a) Search the emulsion for track entries;

b) Determine whether the entry point is isolated or part of a track;

c) Determine directional properties of tracks emanating from an entry point;

d) Trace along tracks until a termination or vertex is encountered. A vertex is defined as a point along a track where splitting occurs; this also includes degenerate splitting, i.e. no splits at all.

e) Determine whether a termination along a track is either an end point of the track or a vertex; if a vertex is detected, the emanating tracks must be classified according to their directional properties for subsequent track tracing;

f) Organize detected entry points, tracks, vertices, and terminations in a storage format which enables the assemblage of several frames. Nuclear emulsions are generally stacked in frames, frames are examined individually, so that the information retrieved from a frame must be coordinated with information retrieved from upper and lower frames.

Figure (III-1) depicts a conceptual block diagram of the optically scaled breadboard model.

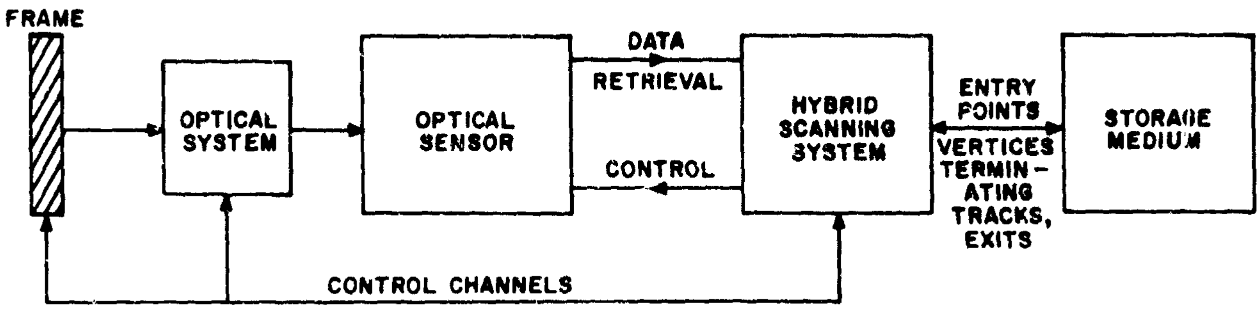


FIGURE (III-1). CONCEPTUALIZATION OF TRACK TRACING

The optical system comprises the collection of lenses which project real images of events, from the nuclear emulsion onto the face of an optical sensor. These images must represent with minimal distortion, the real events within the emulsion. The breadboard model used a simple Cooke triplet with 1:1 magnification and a .375 inch depth of focus.

The optical sensor produces electronic signals from images, projected onto its face, of events or segments of events within the nuclear emulsion. These electronic signals represent the observed measurements taken from the nuclear emulsion -- figuratively, it is the "eye" of the track tracing system. Our breadboard simulation used an electrostatically focused vidicon as the optical sensor.

The means for controlling the relative movement of the frame with respect to the optical system and sensor as well as for movement along an optical axis (i.e. an axis along which optical properties are varied for depth perception) was accomplished by servomechanisms.

A hybrid computer was selected as the medium for implementing the breadboard model of the system because it combined continuous and sequential operations along with storage of data -- precisely the requirements of an emulsion scanning system.

The analog sections provide for the generation of scanning regimes, data retrieval from the vidicon, and generation of control functions for the servomechanisms.

The digital section serves as an executive to coordinate the scanning tasks, to provide for information retrieval, and its transformation to a data base compatible with the assemblage of data from the many frames of a stack.

The implementation for the tasks listed at the beginning of this section, were combined within our optically-scaled breadboard model for a nuclear emulsion track tracing system. The remainder of this report treats the scanning concepts applied to this model and their implementation and the implementation of a data base for the assemblage of data from frames of a stack.

III.3.0 CHARACTERISTICS OF THE OPTICALLY SCALED BREADBOARD MODEL

During the early phase of this study numerous track following and vertex analysis procedures were studied by EAI in an effort to select methods suitable for the analysis of nuclear emulsion stacks by hybrid computer -- vidicon sensor systems. The procedures ultimately selected and incorporated in the logic of software programs subsequently developed are based upon a "vertex-to-vertex" philosophy, administrative control of which resides in a FORTRAN IV main program. Three basic scanning modes are incorporated. They are:

<u>Edge Scanning</u>

For purposes of this study, an edge was defined as the physical boundary of the frame. Track intersections with an edge are located by optically scanning a rectangular grid imposed over the edge surface.

<u>Vertex Analysis</u>

Vertices thought to exist at a point are analyzed by optically constructing a thin spherical shell about the point in question. Confirmation of tracks suggested by "blobs" encountered within the spherical shell itself is accomplished by scanning along a line connecting such blobs with the central point.

<u>Track Following</u>

A track is followed from a starting point with an initial direction by scanning along three directions -- the initial

direction and two adjacent ones -- selecting as a valid direction that angle for which a maximum optical length occurs. This length and angle determine the next position along the track.

The administrative control of the scanning procedures, recording of data and its correlation from a stack, consisting of an arbitrary number of frames, is accomplished by the main program.

III.4.0 THE "VERTEX-TO-VERTEX" PHILOSOPHY

An event, as observed in nuclear emulsions, is a collection of verticles and links. These links, in general, are curved lines as opposed to straight lines, so that a characterization of an event should include vertices and links, coupled with a measure of curvature for the links connecting the vertices.

The events, based on the characterization above, can be recorded in a compact form which utilizes the vertex locations and their multiplicity along with the directional properties of the links connecting the vertices.

This characterization then defines sequences within the track tracing process, namely:

1) Detection of track entries -- treated as vertices.
2) The directional properties of links emanating from a vertex. The number of emanating links is termed the multiplicity of the vertex.

3) The tracing of links emanating from one vertex to another vertex, which may have multiplicity zero (a termination point, or a multiplicity of two or greater (a splitting of the track indicating disintegration phenomena).

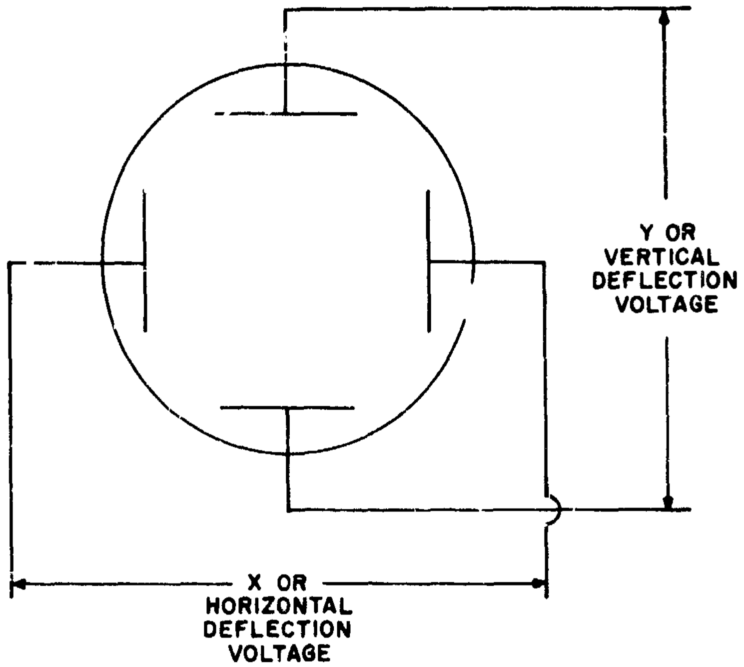
These regimes were incorporated into the software for the optically scaled breadboard model. The administrative control for coordinating the interaction of these regimes resides within the main program. These regimes collectively comprise what we term the "vertex-to-vertex" philosophy of the breadboard model.

III.5.0 THE VIDICON-OPTICS SUBSYSTEM

The vidicon is a photoconductive device which generates electrical current proportional to the light intensity incident on its face -- a photoconductive layer of phosphor. An image projected into the face of the vidicon can be transmitted electrically by exploring the face in a systematic manner and transmitting at each instant the generated current. The result of such a process is to produce a current that varies with time in accordance with the light intensity of successive elements of the image projected into the vidicon's face. This process of exploring an image to obtain a current that varies with time in accordance with the light intensity of successive areas of the image is called scanning.

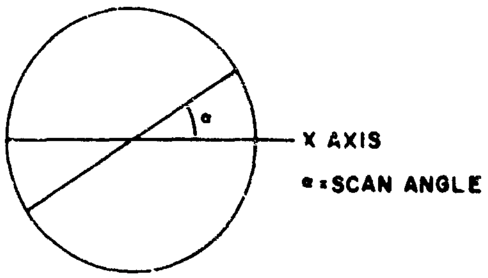
The scanning of the vidicon's face is accomplished by sweeping an electron beam across the photoconductive layer of the vidicon's face by means of electrostatic deflection voltages.

Schematically, the vidicon face is represented below



The deflection voltages control the position of the electron beam's point of impact with the photoconductive layer. The current produced by the photoconductive layer is proportional to the light intensity incident in the layer at the point of impact for the electron beam.

The scanning mode used for the breadboard model vidicon is a radial scan with a variable radius and variable angle. Schematically,



The angle α is measured relative to a horizontal axis. This scanning procedure was realized by generating the x and y deflection voltages in the form:

$$X_{volts} = (R - R_V) \cos_\alpha$$

$$Y_{volts} = (R - R_V) \sin_\alpha$$

with R_V voltage representing the radius of the vidicon face, and R a swept voltage varying from zero to $2R_V$ every 2 milliseconds.

The analog implementation of this scanning procedure appears in Figure (III-2).

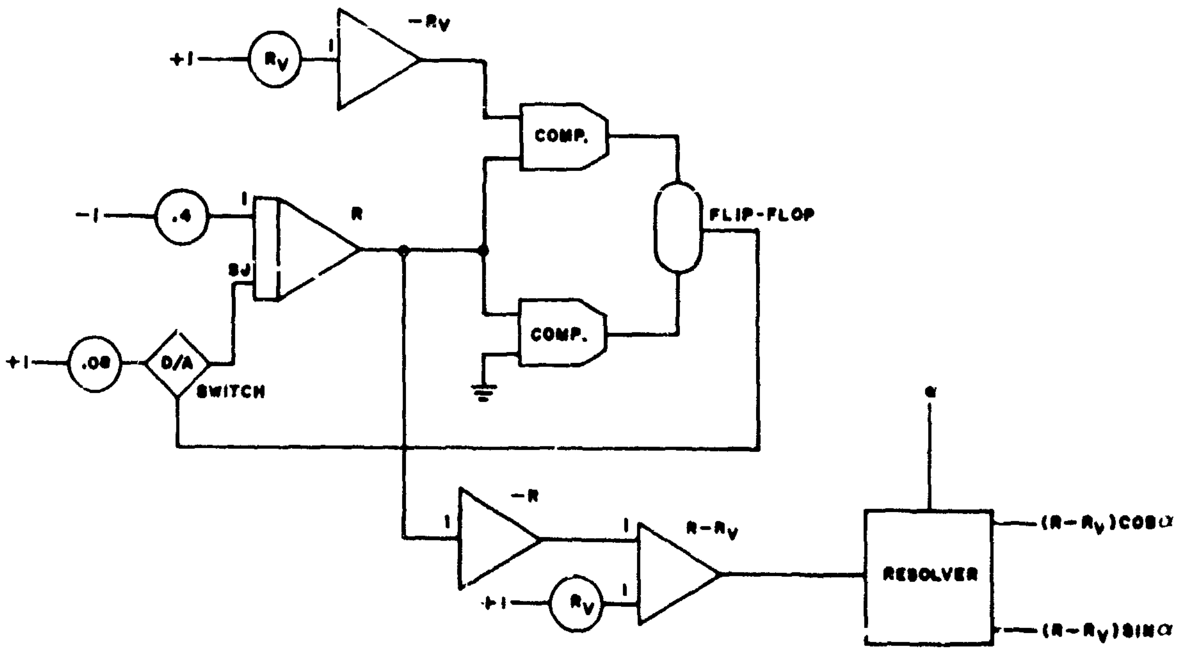


FIGURE (III-2). RADIAL SCAN VOLTAGE GENERATION

The waveform of R produced by the above oscillator is shown in Figure (III-3)

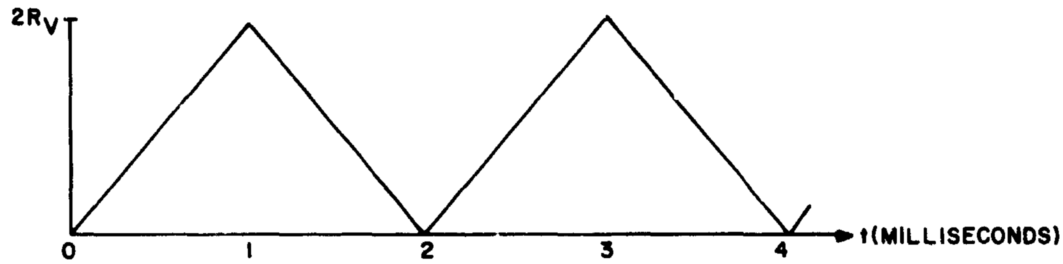
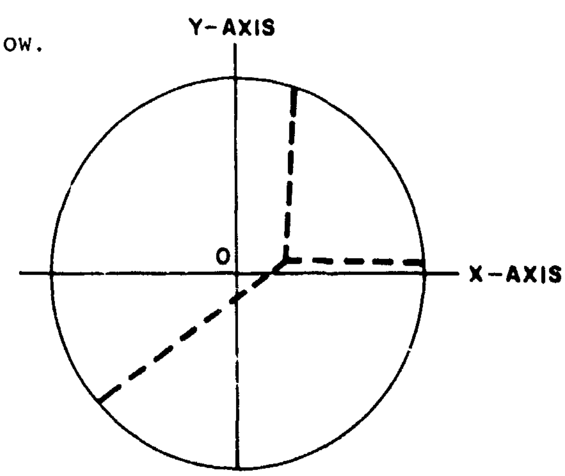


FIGURE (III-3). WAVE SHAPE OF THE RADIAL SCAN VOLTAGE

Information is retrieved during the rising slope of the radial voltage, on the descending slope the vidicon is blanked to reduce the average current produced by the vidicon -- a protective measure.

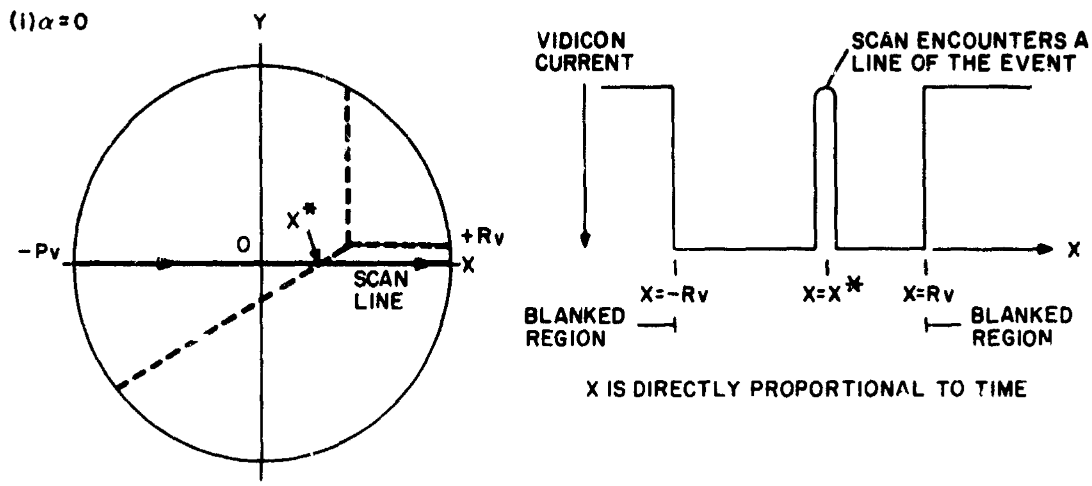
The Vidicon Output

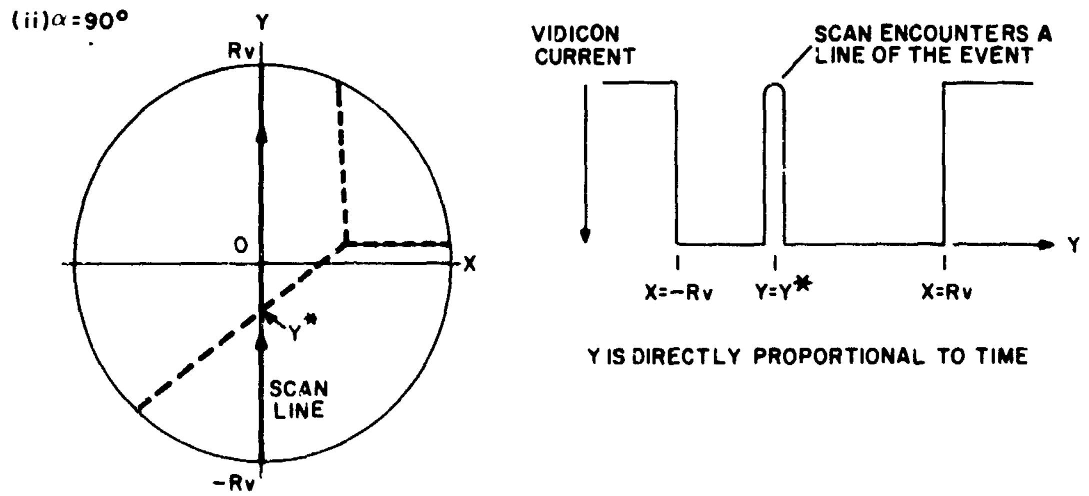
We consider the vidicon face to be a disk onto which images of segments of events are projected, as shown below.

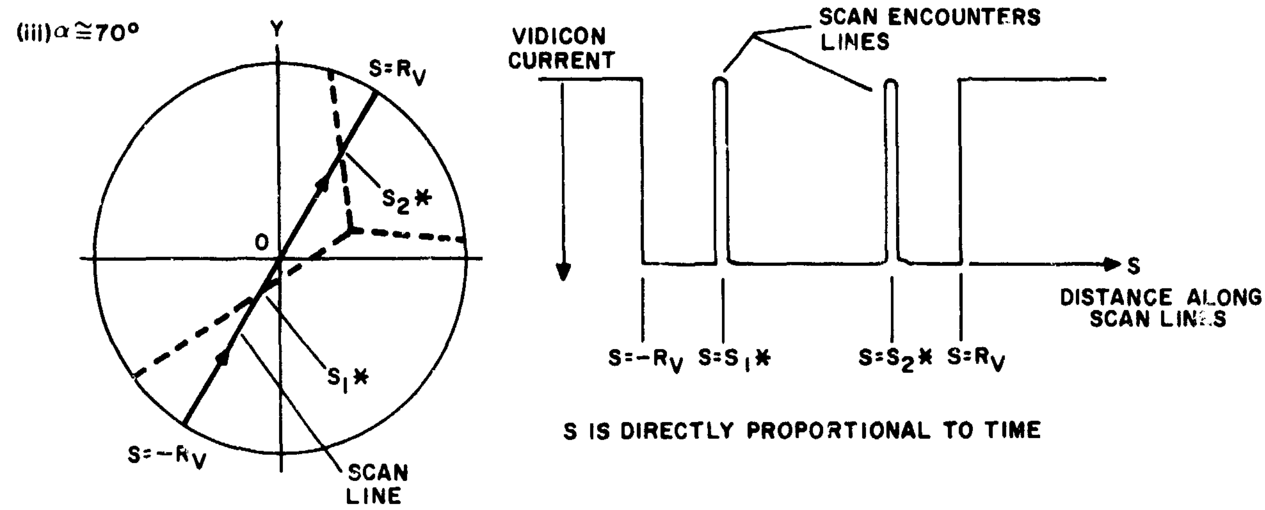


Establishing a coordinate axes system with an origin at the center of the disk fixes a reference frame for projected images of events.

For each scan angle, α, the vidicon scan of its face -- for the projected image above -- produces a current proportional to the intensity of light incident on its face. Typical vidicon outputs are shown below:







Examination of the above scan shows that when the scan encounters a dark region, such as a portion of a line, the light intensity decreases causing the current to drop to zero producing a pulse shaped output. The location of these pulses relative to the origin of the established coordinates is determined from the time at which the scan encountered the time segment.

III.6.0 EDGE SCANNING IN THE BREADBOARD MODEL

Our study assumed that the stack of frames was shielded from above and below and that all observable events begin on the sides of the stack. This assumption covers the largest class of expected events. These are, however, events which are visible only within the interior of a stack, e.g. non-interacting primary particles which decay within the stack and produce interacting particles, that is, non-interacting particles are not visible as tracks.

With this assumption that events begin on the edge of a frame in the stack the track entry point is located by imposing a rectangular grid of cells over the edge of a frame and determining which cells have been

exposed. That is, the entry point is not really a point but a region of exposed grams of emulsion -- a blob -- on the side of a frame. The exposed region must be enclosed by a closed curve and its centroid used as the point of entry of a track.

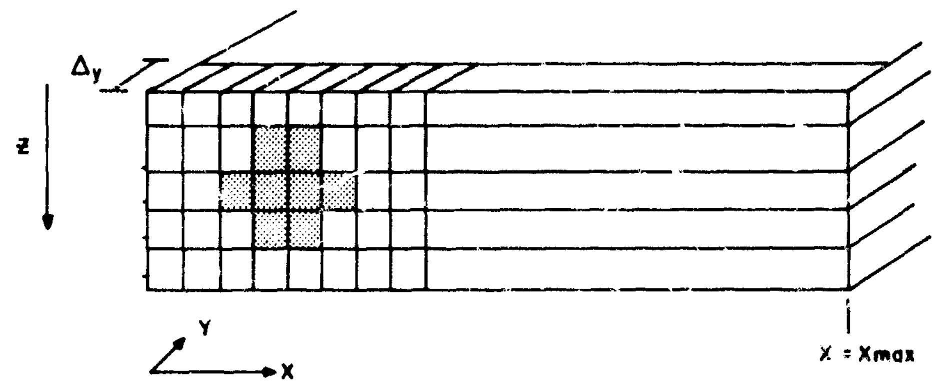


FIGURE (III-4). THE BLOB CONFIGURATION ON THE EDGE OF A FRAME

Figure (III-4) depicts the edge of a frame with a rectangular grid imposed. The rectangular grid is generated by the points:

$x_i = i\Delta x$	$i = 0,1,2,\ldots,N$
$z_i = i\Delta z$	$i = 0,1,2,\ldots,M$

The z_i points refer to the plane of focus of the optical system with Δz its depth of focus. Δy corresponds to the width of the electron beam. Δx is the scan window for the vidicon face, e.g., $\Delta x = 2R_V$ for a coarse grid, $\Delta x = K2R_V$, $K<1$ for a finer grid.

A blob is indicated by the darkened cells within the grid. Generally, the blobs do not occupy cells in so orderly a manner, but for our purposes this sketch does not impose any difficulties.

For the breadboard model, the track entries were represented as the intersection of line segments from events and the physical boundary of the frame face.

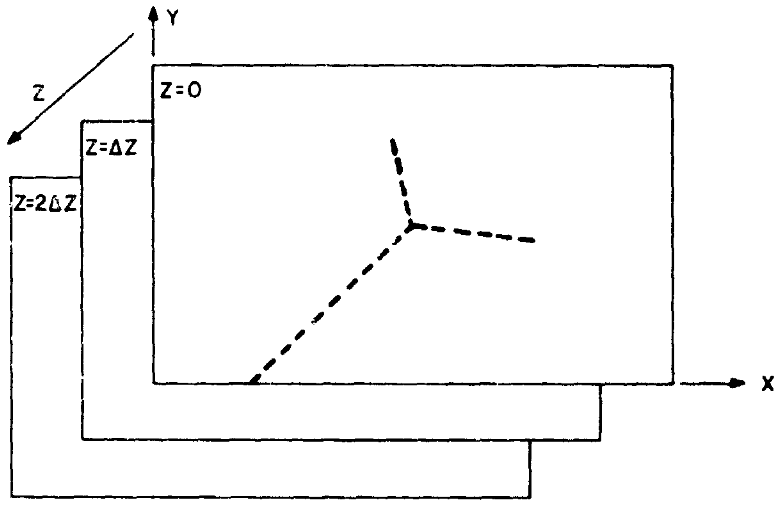


FIGURE (III-5). THE BREADBOARD MODEL IMPLEMENTATION OF A FRAME

Figure (III-5) depicts the implementation of a frame in the breadboard model. Three levels of z were used -- due to limitations in physical size and optical parameters of our components. Each level, or z-phase of a frame comprised a view graph slide with events represented by thin strips of tape. Three dimensional tracks were represented by taping projections in each z-level.

The edge scanning proceeded by moving the frame relative to the vidicon optics assembly.

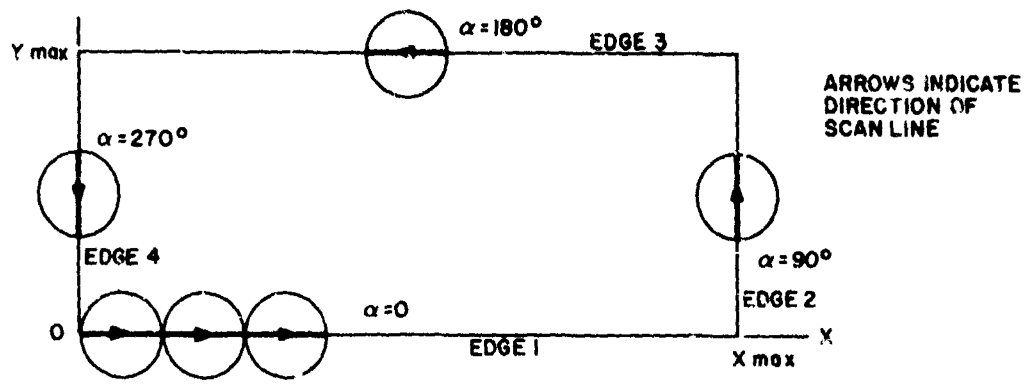


FIGURE (III-6). THE EDGE SCAN TECHNIQUE

The rectangular shape in Figure (III-6) represents the physical boundaries of a frame. The circles imposed along these edges represent positions of the vidicon optics assembly relative to the frames edges. The edges of the frame are numbered 1,2,3, and 4; respectively; for

$x = 0,\ 0 \le x \le x_{max};\ x = x_{max},\ 0 \le y \le y_{max};\ y = y_{max},\ 0 \le x \le x_{max};\ x = 0,\ 0 \le y \le y_{max}.$

This edge scan program is illustrated by Figure (III-7)

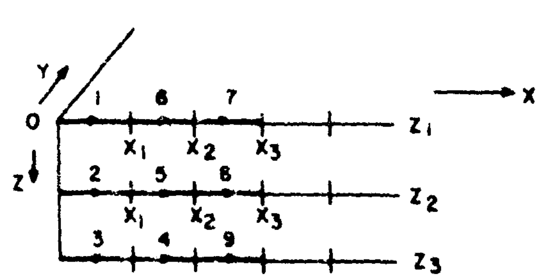
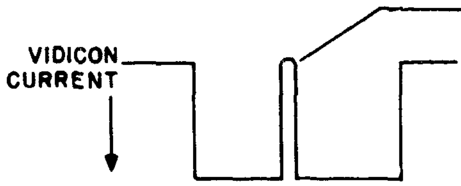


FIGURE (III-7). ILLUSTRATION OF EDGE SCAN PRODEDURE

The electron beam when scanned across the vidicon face with $\alpha=0$ produces a pulse on encountering the line segment.



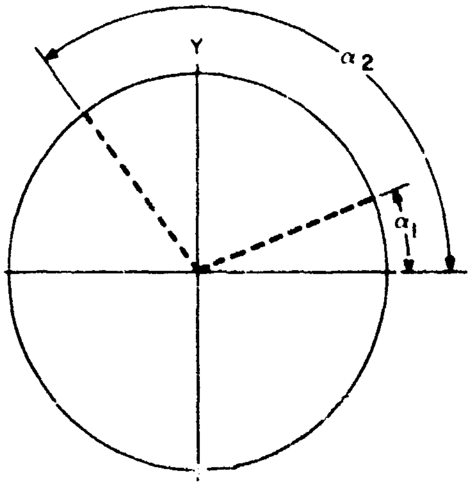
pulse indicating intersection of the scanning beam with the projected image of the line segment.

The time of occurrence of this pulse, relative to the start, determines the coordinate of this entry point in the reference frame of the vidicon face. Upon detection of an entry point, the main program proceeds to a vertex analysis. The vertex analysis determines whether or not tracks emanate from the point and if so, their number and respective direction angles.

III.7.0 VERTEX ANALYSIS IN THE BREADBOARD MODEL

Once a vertex, such as a track entry point, is detected, the vertex is analyzed, or processed, to identify emanating lines.

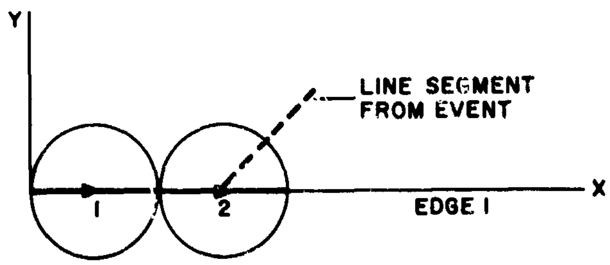
The process begins by positioning the center of the vidicon optics assembly over the vertex point.



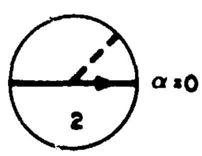
Assuming an origin $O(x=0, y=0, z=0)$, the first cell scanned is cell 1 $(z_1, 0 \leq x \leq x_1)$, if a track entry is detected, we proceed to a vertex analysis of the entry point, if not, we scan cell $2(z_2, 0 \leq x \leq x_1)$ by moving the vidicon optics assembly along the z axis, if no track entries are detected we scan cell $3(z_3, 0 \leq x \leq x_1)$, again if no track entry appears we scan cell $4(z_3, x_1 \leq x \leq x_2)$ by moving the vidicon assembly to a new starting position along the frame edge $(x=x_1)$, and proceed to scan, provided no entries are detected, cells 5,6,..., etc.

Detection of an Entry Point

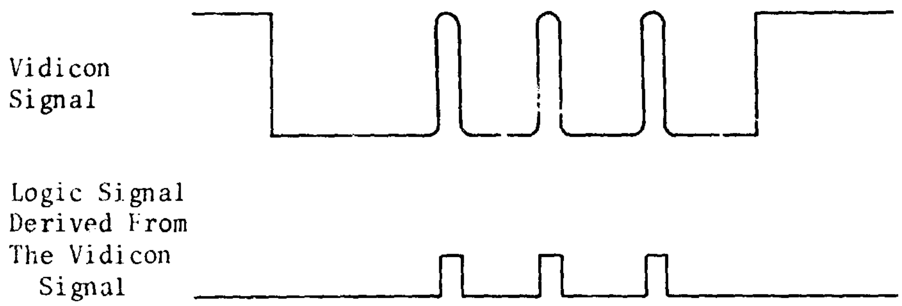
An entry point is defined as the intersection of the frame boundary and a line segment of an event. As we scan along the edge of a frame, as shown above, images are projected onto the vidicon face.



The first position, of the assembly, circle 1, will not have a line segment projected onto its face. The second position, circle 2, of the assembly will have the image of the line segment projected onto its face.

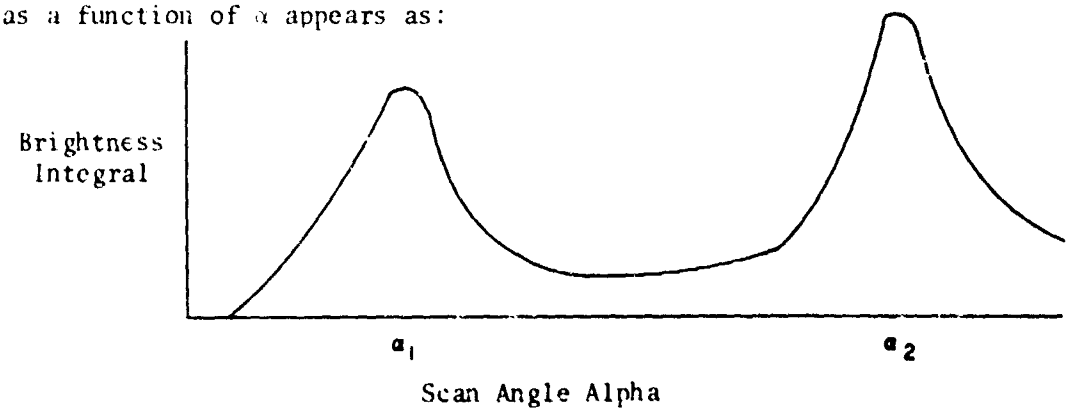


An image of the vertex point and its neighboring area is projected onto the face of the vidicon optics assembly. The vidicon face is scanned for successive α scan angles, for each such scan angles, for each such scan angle, an integral derived from the vidicon signal is computer, referred to as the Brightness Integral.



With a vidicon signal as shown above, a logic signal is derived by comparison against a threshold, during the data retrieval portion of the radial scan. This logic signal controls the operate mode of an integrator with constant input. So that the integrator value, at the end of a scan, represents a measure of coincidence for the scan with a line segment projected on the vidicon face.

If we graphically represent the results of this procedure, the brightness integral as a function of α appears as:



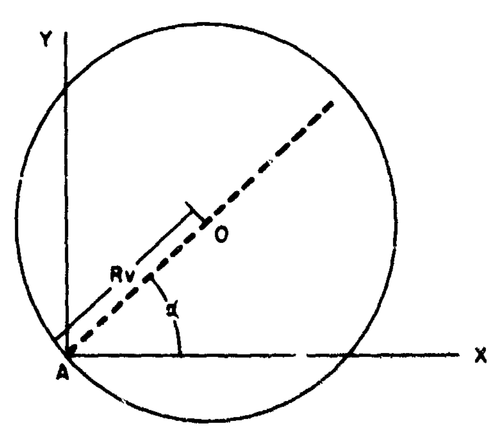
Peaks occur at the scan angles α_1 and α_2 along which the projected image has line segments. The brightness integral also serves to establish, when compared against a threshold value, the existence of a line segment. These peaks are detected by the main program -- at present the program is capable of recognizing as many as five peaks -- but this can be extended to include greater numbers.

Each of the detected line segments emanating from a vertex point is then traced along its path until a new vertex -- either a termination, or a splitting point -- appears, at which time the vertex analysis procedure is repeated.

III.8.0 TRACK TRACING ON THE BREADBOARD MODEL

The track tracing regime occurs after the disclosure of non-zero direction angles for emanating tracks of a vertex. The track trace is initialized with a vertex point and a direction angle. The track tracing continues until a new vertex is encountered.

In the diagram below, A is the vertex point -- the starting point of the track tracing process. O is the center of the vidicon.

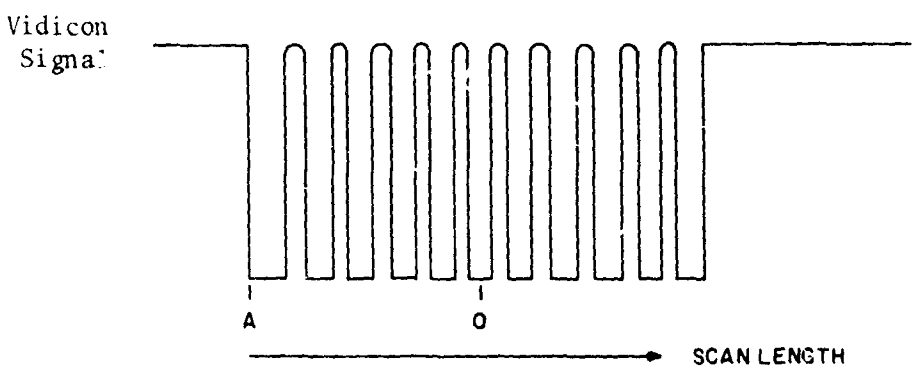


The vidicon optics assembly is positioned at the point O, computing the assembly's coordinates as

$$x_V = x_A + R_V \cos\alpha$$
$$y_V = y_A + R_V \sin\alpha$$

The heavy dashed line in the diagram represents the projected image of a line segment emanating from the vertex of point A with direction angle α

The vidicon scans from point A along the angle α. The corresponding vidicon signal appears as a series of pulses, each pulse representing an encounter of the electron beam with a dashed section of the line.



That scan length for whcih pulses no longer appear is recorded as the maximum travel, R_{scan}, from point A along the direction angle α of coincidence between the line segment and scan. For the case, shown in the above figure, $R_{scan} = 2R_V$.

For each starting point of a known line segment, such as A, three successive angular scans take place; $\alpha-\Delta\alpha$, α, $\alpha+\Delta\alpha$ with $\Delta\alpha$ an incremental change

in α (approximately 2 degrees). Each of these scans, starting at A, result in a maximum length $R_{scan}(\alpha)$. That α for which $R_{scan}(\alpha)$ is maximized is defined as the corrected direction angle for the line segment. This α is used to reposition the vidicon assembly on a new starting point and the procedure repeated until a new vertex appears - coincidence of the track and scan no longer occurs.

A minimal measure of coincidence, or scan length defined as a threshold, serves to detect the occurrence of a new vertex whenever coincidence falls below this threshold value.

The repositioning is computer as

$$x_V = x_A + R_{max} * \cos\alpha_{max}$$

$$y_V = y_A + R_{max} * \sin\alpha_{max}$$

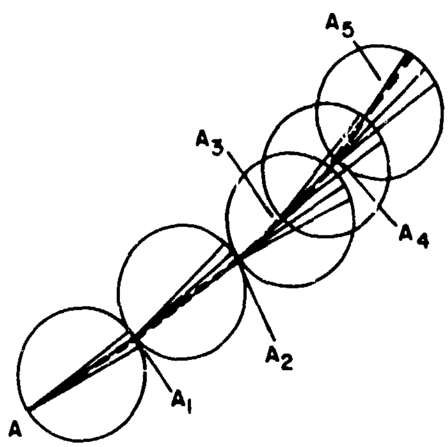


FIGURE (III-8)

Figure (III-8) illustrates successive stages of this process for track tracing an event from point A to point A_5.

III.9.0 THE MAIN PROGRAM - COORDINATION OF THE SCANNING MODES

To demonstrate the data handling capability of the main program digital simulations of the analog scanning mode for the system were constructed. Utilizing tab card descriptions of the tracks, the program 1) Identifies track entries through frame edges, 2) Traces events from vertex to vertex, 3) Assembles event related data requiring scanning of more than one frame, and 4) Summarizes results of completed analysis and produces both card and printed copy tabulations of all vertex coordinates associated with each event identified.

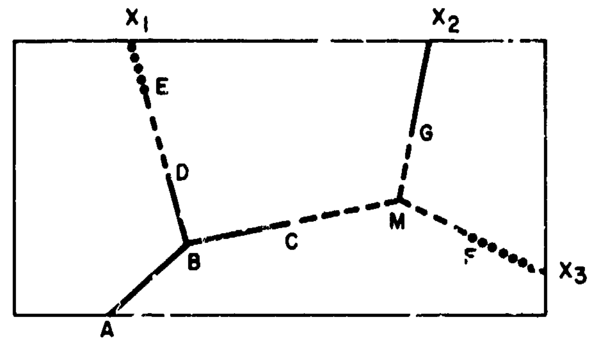
The program utilizes a single working tape and instructs an operator to mount emulsion frames (card deck simulations) as called for by the logical assembly of acquired data.

Presented below is a description of the operation of the main program. The program will be described by following its operation on a hypothetical event which embodies most possible complexities.

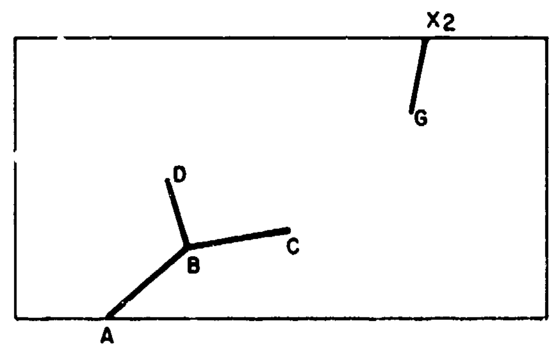
For our hypothetical case we treat an event which covers three frames of a stack. The event is represented below by its projections into each of the three frames and by a composite view of projections for all three frames.

The event, as sketched below represents a particle track entering frame 1, at A. The track travels to B, a vertex of order 2, where a splitting occurs forming two new tracks. One of these emanating tracks from the vertex at B travels to D where it descends into the second frame. This track passes through the second frame, D to E and descends to the third frame where it

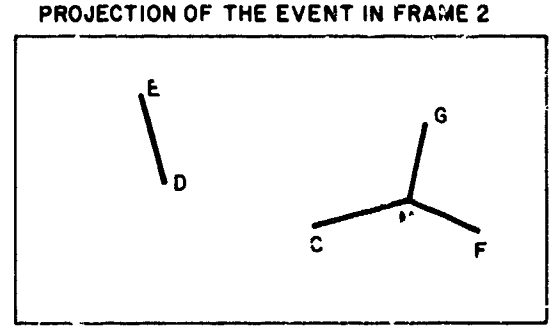
1) A COMPOSITE VIEW OF PROJECTIONS IN FRAMES 1, 2, AND 3



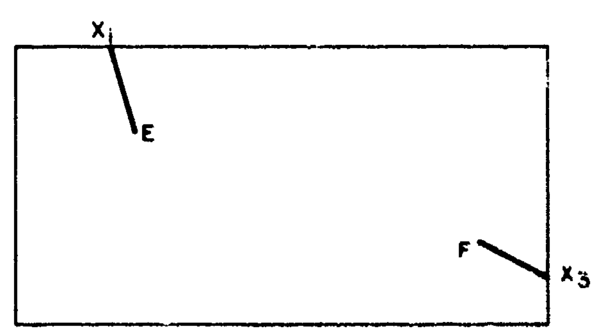
2) PROJECTIONS OF THE EVENT IN FRAME 1



3) PROJECTION OF THE EVENT IN FRAME 2



4) PROJECTION OF THE EVENT IN FRAME 3



exits at edge 3, E to X_1. The second emanating track from B travels to C where it then descends into the second frame. In the second frame the track travels from C to M, a vertex of order 2. Two new tracks are formed. The first of these tracks travels to F where it then descends into the third frame and exits at edge 2, F to X_3. The second track leaving M travels to G where it then ascends to frame 1 and exits at edge 3, G to X_2.

Flow diagrams detailing the operation of the program are shown in Gigures (III-9) through (III-20).

The main program begins with initialization of arrays and variables which indicate the status of the tracing procedure. As our hypothetical event is traced we will encounter these arrays and variables and, therefore, defer their definitions.

The flow diagram of Figure (III-9), illustrates the identification of a frame (for our first frame, FRAME=1), the initialization of pertinent edge scan arrays, and variables, and a test for determining whether any interframe tracks are to be considered in the frame under study. At this point we have not as yet discovered any interframe transfers, so that we proceed to test for completion of the stack edge scan, Figure (III-10).

STKFLG and EDGFLG are integer variables with two possible values -- 1 and 0. STKFLG=1 signifies that the edge scanning of all frames in the stack is incomplete, while STKFLG=0 signifies completion. EDGFLG=1 signifies edge scanning of frame selections in normal frame number sequence. EDGFLG=0 signifies that the normal edge scanning of a frame selection is interrupted due to an excessive number of interframe track references.

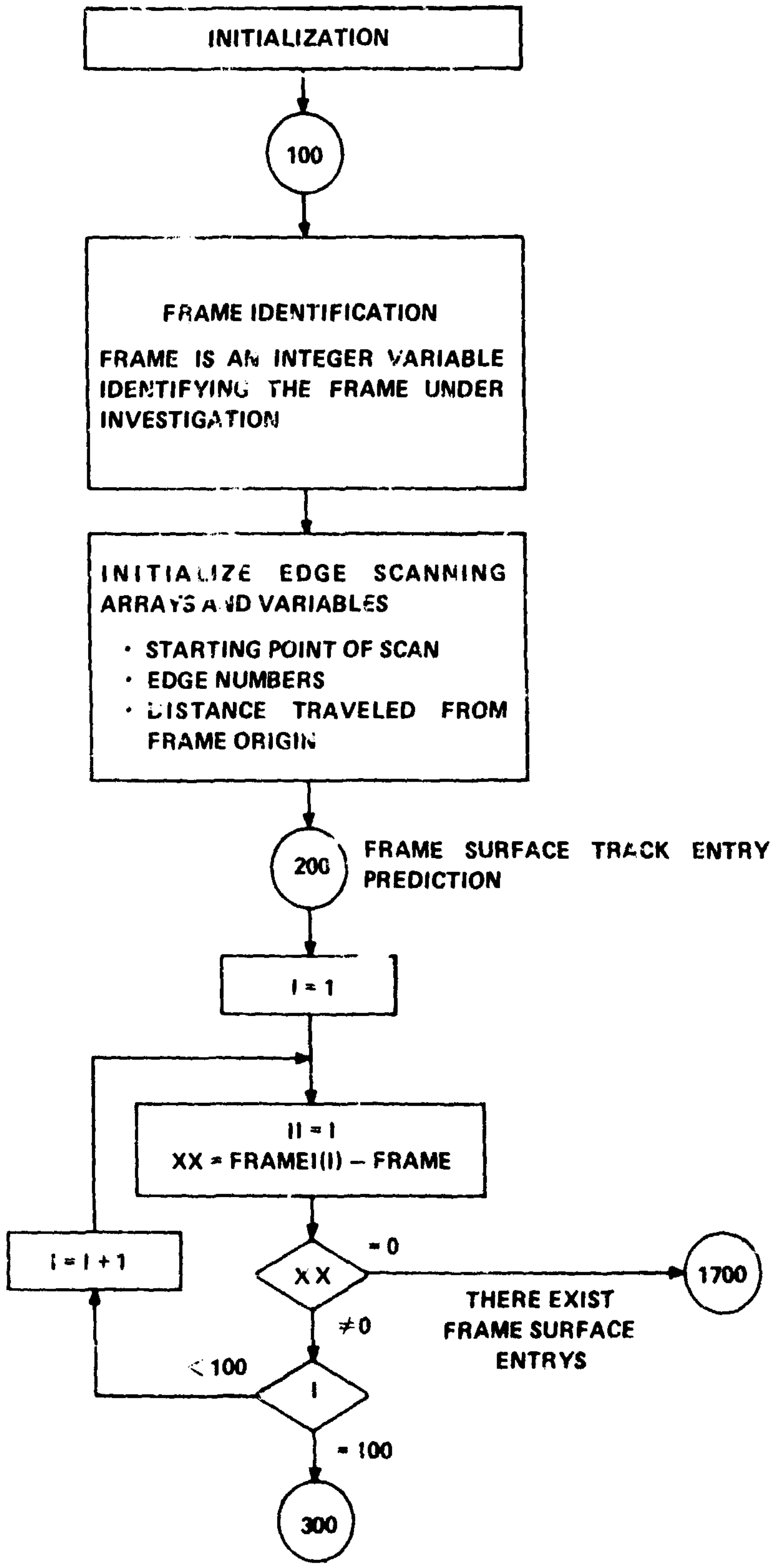


FIGURE (VII-9) - INITIALIZATION PROCEDURES

A starting point for the edge scan of the selected frame is defined -- initially we start at (0,0,0). The edge scanning routine is entered and a possible track entry point is retrieved; namely, point A on edge 1 of our hypothetical event, i.e.

$$XO(1) = X_A$$

$$YO(1) = Y_A$$

$$ZO(1) = Z_A$$

This entry point is compared with the starting point (0,0,0). $X_A+Y_A+Z_A=0$, implies a complete traversal of the frame edges and, therefore, a return to the origin.

This entry point then serves as the starting point for subsequent edge scanning, so that we redefine XS, YS, and ZS accordingly,

$$XS = X_A$$

$$YS = Y_A$$

$$ZS = Z_A$$

In Figure (III-11), the entry point is compared against previously defined exit points -- it may happen that this entry point is actually an exit point for a previously traced event. Initially the arrays XE, YE, and ZE are defined with zero values so that the program flow proceeds to the track entry confirmation procedure.

The distance from the origin, traveling along the edges of the frame, is computed, i.e.

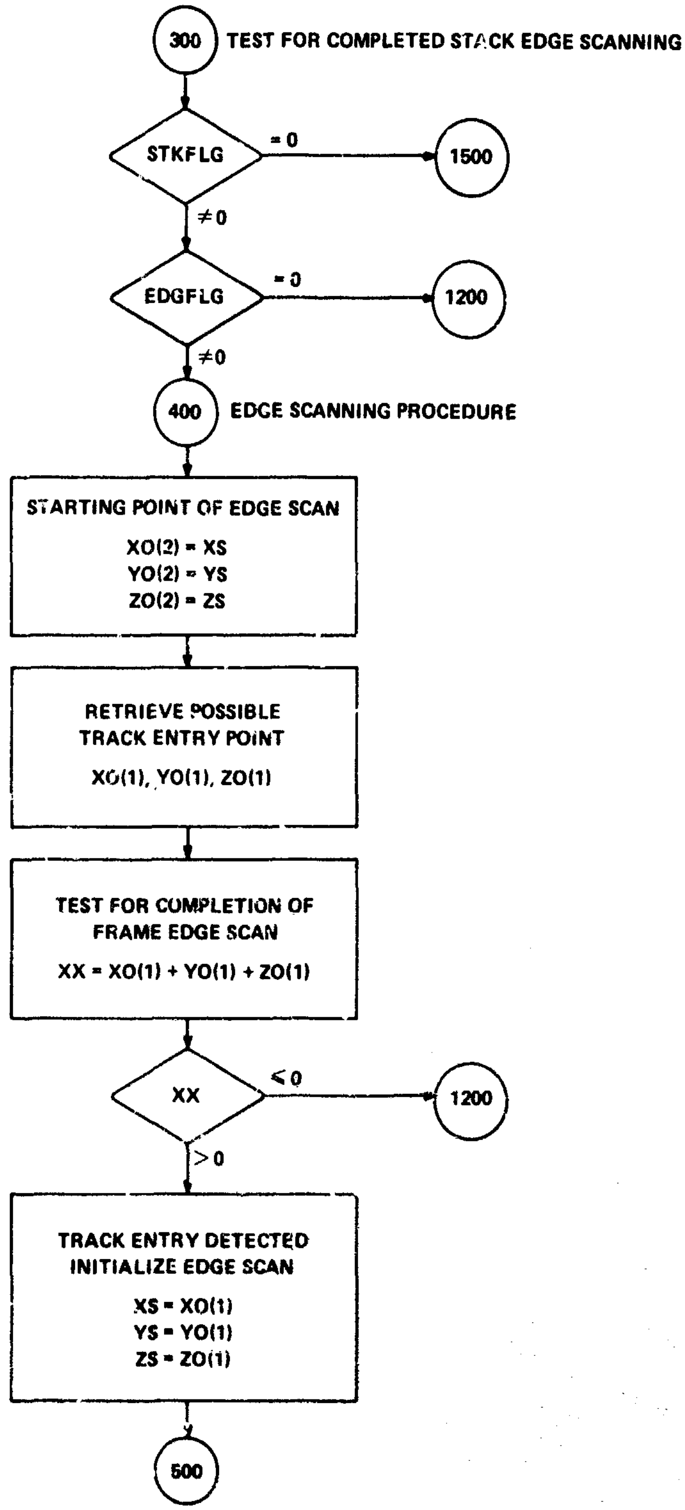
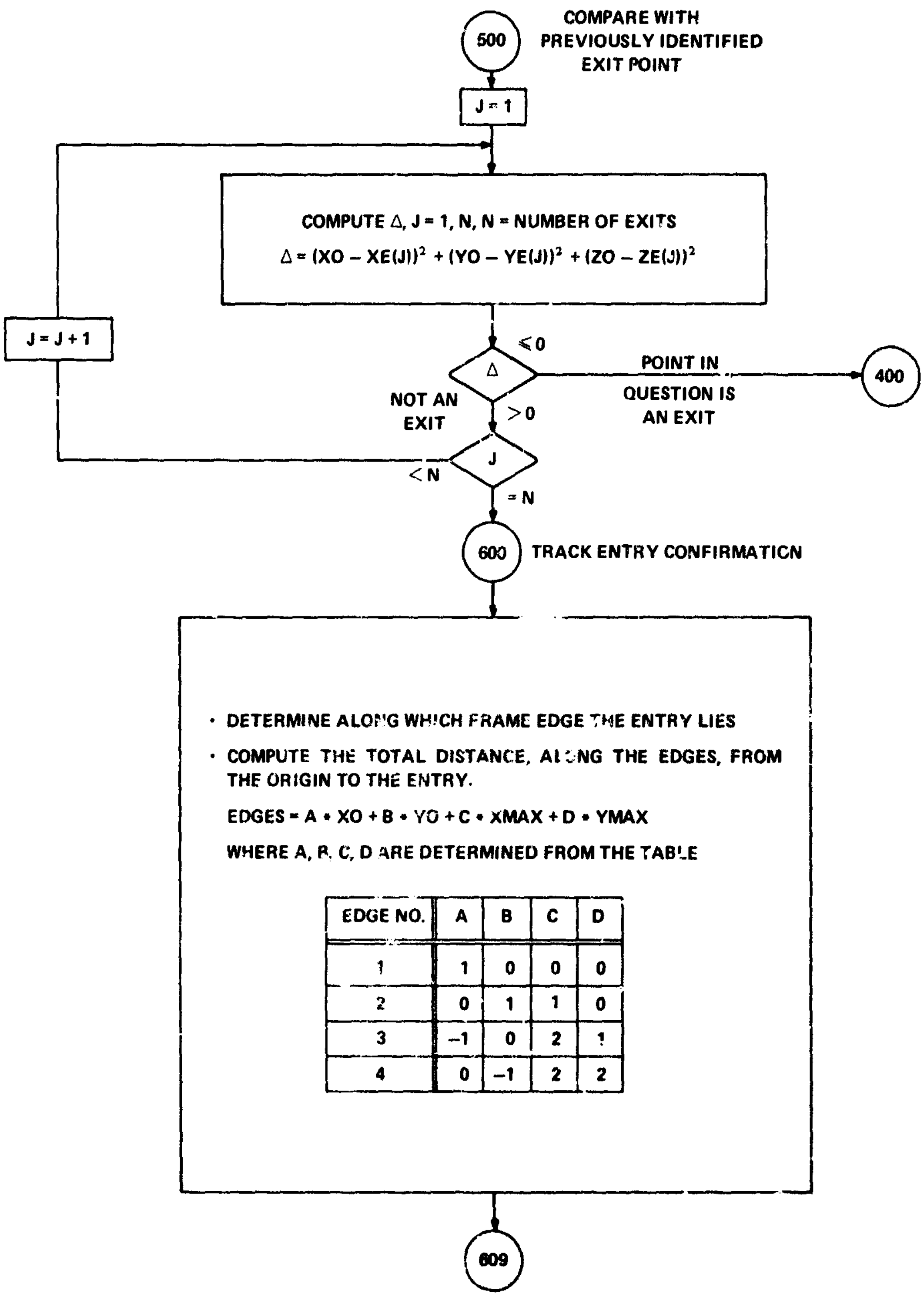


Figure (III-10) Edge Scan Procedure



EDGE NO.	A	B	C	D
1	1	0	0	0
2	0	1	1	0
3	–1	0	2	1
4	0	–1	2	2

Figure (III-11) Exit Comparison and Confirmation

$EDGES = X_A$, since the edge number equals one.

The entry point, considered a vertex, is analyzed to determine its multiplicity or the number of emanating tracks, from A. The vertex analysis is performed by subroutine VERTEX. The arrays ALPHA, and SINBT are returned to the main program; ALPHA contains direction angles in the plane of the frame, SINBT contains the sine of departure angles along the depth, or z axis, of the frame. The first non-zero ALPHA(1) is recorded as an event, intially EVENTS=0, therefore, following Figure (III-12), EVENT=1. ORDER is an array, initialized with zero values, which contains the order -- the number of emanating tracks -- of a designated vertex, i.e. A is designated vertex 1.

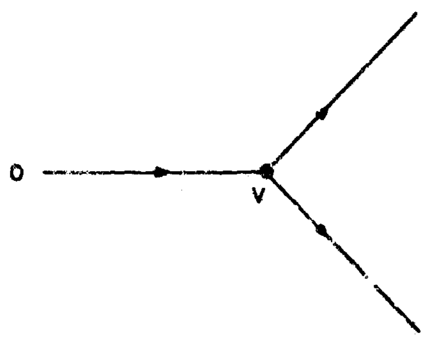
Following along, Figure (III-13), the vertex A is stored in the array COORDT. STAR is the integer variable denoting vertex numbers of an event.

With STAR = 1, we have

$COORDT(1,1) = X_A$

$COORDT(1,2) = Y_A$

$COORDT(1,3) = Z_A$



V IS ENCOUNTERED BY THE TRACK OV.

The order of the vertex, A, is determined by testing the ALPHA array for non-zero values which do not complement the original track entering A. In general, an interior vertex is encountered along a track, when determining its order the original track should not contribute. For example, in the sketch below.

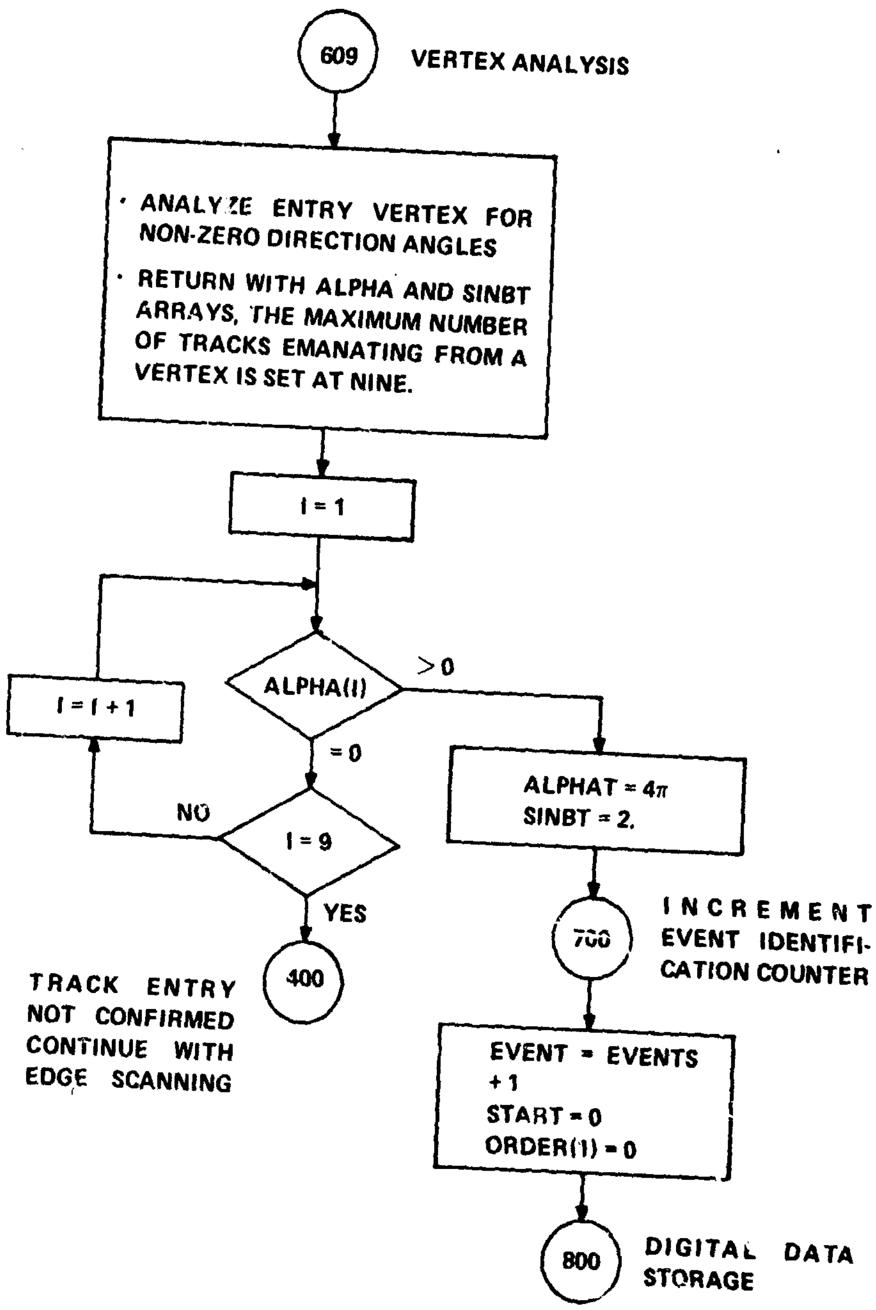


FIGURE (III-12) - VERTEX ANALYSIS

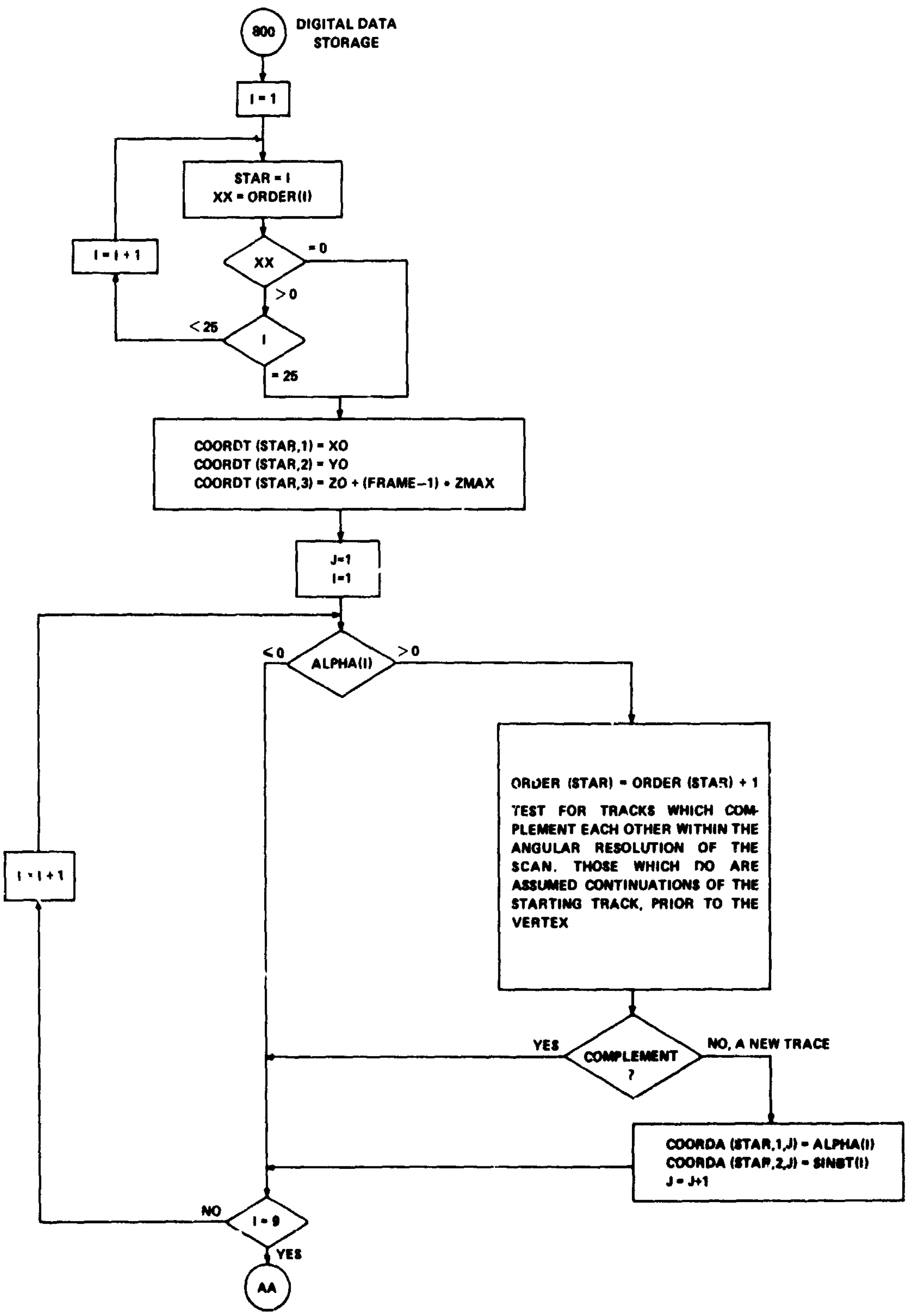


Figure (III-13) Digital Storage Format

V is an interior vertex -- internal to the frame boundaries -- the order of V is 2 not 3.

The order of vertex A is 1, since only one track leaves A. The values of ALPHA(1) and SINBT(1) are stored in COORDA,

$$COORDA(1,1,1) = \alpha_A$$

$$COORDA(1,2,1) = \sin\beta_A$$

where α_A and β_A are respectively the planar direction and planar departure angles for the track leaving A.

The first subscript in COORDA is the vertex number, STAR, the second subscript designates either planar direction or departure, and the third subscript identifies the amanating track, presently J=1.

In Figure (III-14), the vertex is examined for a point of exit, if it is an exit point it is recorded in the arrays XE, YE, and ZE. The subscript S represents a running count of exit points. The vertex, A, is then tested for coincidence -- within a tolerance -- with the preceding vertex. If coincidence occurs, the present vertex is identified as the preceding one and any emanating tracks are added to its order. If coincidence does not occur, as is the present instance, a check is made for an interframe transfer, i.e. a penetration into either an upper or lower frame. FRAME 1 is an array which lists the frames into which tracks penetrate from the event, initially FRAME 1 is set to zero.

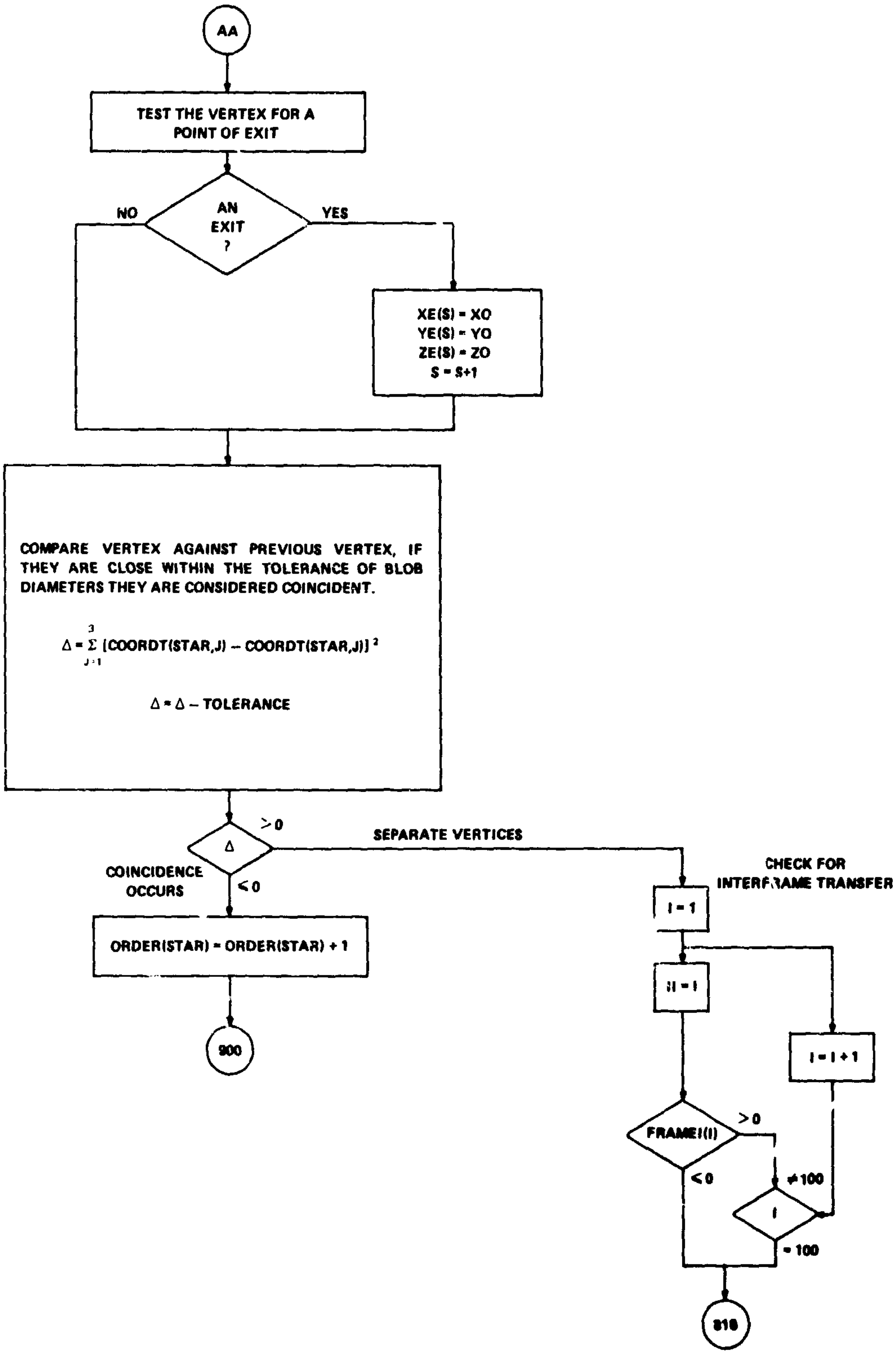


Figure (III-14) Test for Exit and Coincidence

If an interframe transfer does occur, it is catalogued in the arrays FRAME 1, COORD 1, EVENT 1, STARNO, and the variable FRMTRN. In the present case, an interframe transfer has not yet occurred.

The next step in the scanning process is an evaluation of the event status, Fig. 5 (III-15) and (III-16). The status of an event is evaluated by comparing the number of tracks traced from a vertex with its order. If the number of tracks is less than the order of the vertex, this indicates that the tracing of tracks emanating from the vertex is incomplete, and therefore the remaining tracks are traced. The integer variable START is an identifying number for a vertex with remaining tracks to be traced. The integer variable COUNT maintains a running index for the number of tracks emanating from the vertex.

For the hypothetical event,

START = 1

COUNT = 1

ALPHAT = COORDA(1,1,COUNT) = α_A

SINBTT = COORDA(1,2,COUNT) = $\sin\beta_A$

$X_o(2) = X_A$

$Y_c(2) = Y_A$

$Z_c(2) = Z_A$

The track leaving vertex A, identified by the value of COUNT, is traced. This track is traced using the track following routine, starting at vertex A with direction angles defined by ALPHAT, and SINBTT. The track is followed until a vertex is reached, presently vertex B. An analysis of

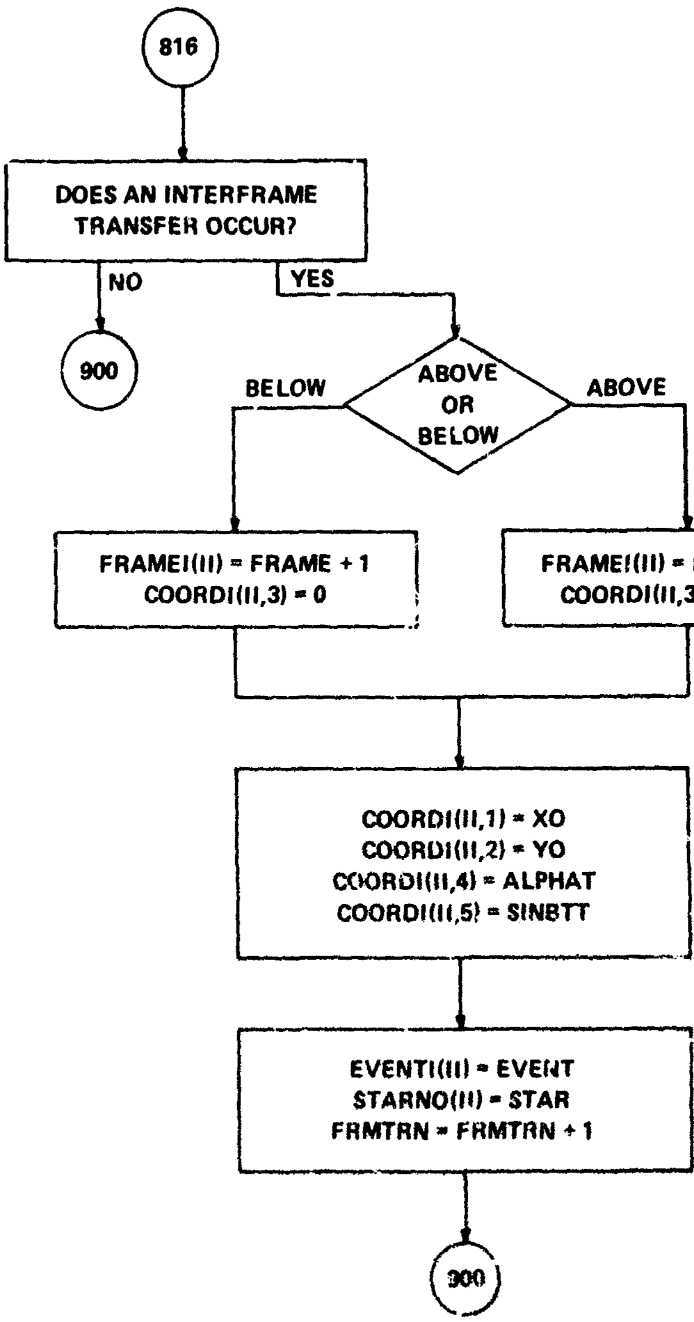


FIGURE (III-15) - INTERFRAME TRANSFER STATUS

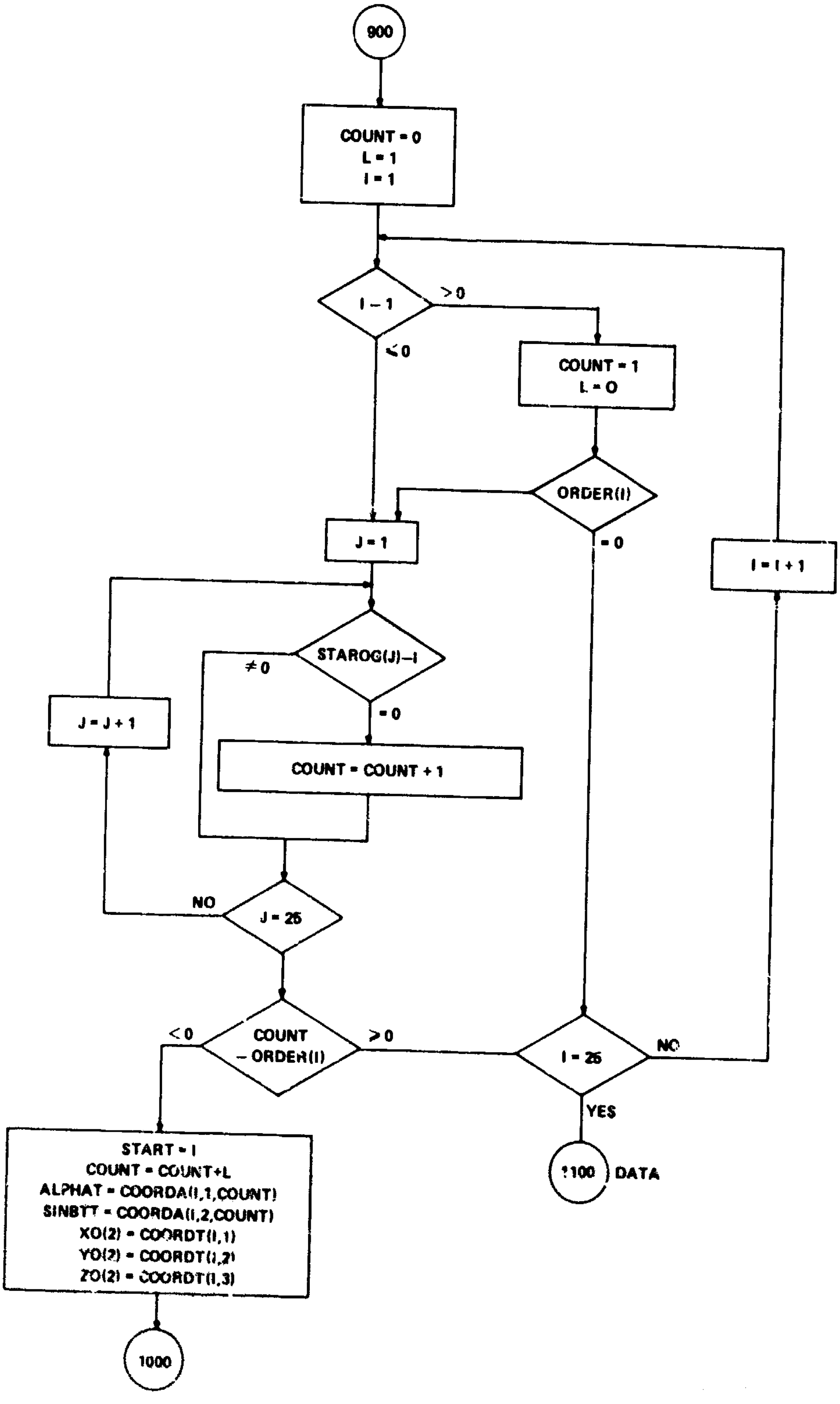


Figure (III-16) Event Scanning Status

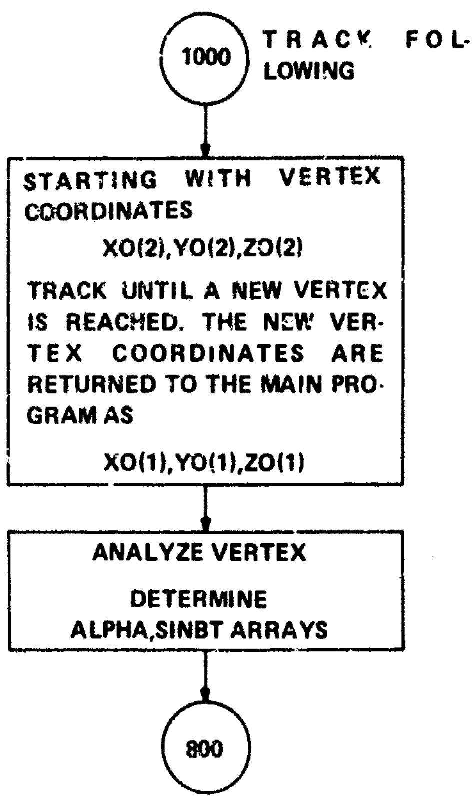


FIGURE (III-17) - TRACK FOLLOWING

this vertex, B, results in the detection of three emanating tracks, defined by:

$$\text{ALPHA}(1) = \alpha_{21} \qquad \text{SINBT}(1) = \sin\beta_{21}$$

$$\text{ALPHA}(2) = \alpha_{22} \qquad \text{SINBT}(2) = \sin\beta_{22}$$

$$\text{ALPHA}(3) = \alpha_{23} \qquad \text{SINBT}(3) = \sin\beta_{23}$$

The third elements of the arrays represent the original track, and will be eliminated during the test for complementation.

Returning to Figure (III-13), vertex B is designated as vertex 2, STAR=2, and its coordinates are stored in the array COORDT.

$$\text{COORDT(STAR,1)} = X_B$$

$$\text{COORDT(STAR,2)} = Y_B$$

$$\text{COORDT(STAR,3)} = Z_B + \text{(FRAME-1)*ZMAX}$$

The order of vertex B is deduced and stored as an element of the ORDER array with subscript STAR.

ORDER(STAR) = 2

The third emanating track, from A, is eliminated during the test for complementation and the COORDA array is computed.

$$\text{COORDA(STAR,1,1)} = \text{ALPHA}(1) = \alpha_{21}$$

$$\text{COORDA(STAR,2,1)} = \text{SINBT}(1) = \sin\beta_{21}$$

$$\text{COORDA(STAR,1,2)} = \text{ALPHA}(2) = \alpha_{22}$$

$$\text{COORDA(STAR,2,2)} = \text{SINBT}(2) = \sin\beta_{22}$$

Continuing with Figure (III-14), vertex B is tested for a point of exit, which it isn't, and then for coincidence with vertex A. Vertices B and A

are not coincident, the program then checks for an interframe transfer, and returns to evaluate the event scanning status, Figure (III-15). Since vertex 1,A, had only one emanating track, which has already been traced, the tracks emanating from vertex 2,B, will now be traced. The first track to be traced is defined for the track following routine by

START = 2

COUNT = 1

ALPHAT = COORDA(2,1,1) = α_{21}

SINBTT = COORDA(2,2,1) = $\sin\beta_{21}$

$X_0(2)$ = COORDT(2,1) = X_B

$Y_0(2)$ = COORDT(2,2) = Y_B

$Z_0(2)$ = COORDT(2,3) = Z_B

The first track leaving B is traced until the vertex at C is reached. This new vertex is analyzed to determine its order and direction angles.

The coordinates of vertex C, vertex 3 (STAR=3), are stored in COORDT.

COORDT(3,1) = x_c

COORDT(3,2) = y_c

COORDT(3,3) = z_c + (FRAME-1)*zMAX

The order of C is 1, since C is a vertex due to the interframe transfer (Frame 1 to Frame 2). The direction angles returned for tracks leaving C complement the entering track direction angles and accordingly, are deleted from the list in COORDA.

The interframe transfer, at vertex C, is recognized, Figure (III-15), and recorded in the appropriate arrays, accordingly;

FRAME1(1) = FRAME+1 = 2

COORD1(1,3) = 0

COORD1(1,1) = X_c

COORD1(1,2) = Z_c

COORD1(1,4) = α_{21}

COORD1(1,5) = $\sin\beta_{21}$

An interframe event counter EVENT1 is listed with

EVENT1(1) = EVENT = 1

i.e., the interframe event counter links the transfer to the event under study. The vertex number of the event is recorded.

STARNO(1) = STAR = 3

The number of frame transfers is updated

FRMTRN = FRMTRN + 1 = 1

and the program returns to evaluate the scanning status of the event.

The evaluation of the event scanning status results in the tracing of the second track leaving vertex B. This track is defined, for the track following routine, by

START = 3

ALPHAT = COORDA(2,1,2) = α_{22}

SINBT = COORDA(2,2,2) = $\sin\beta_{22}$

$X_o(2) = X_B$

$Y_o(2) = Y_B$

$Z_o(2) = Z_B$

The vertex at D is reached by the track following routine; it is then analyzed and recorded as

STAR = 4

COORDT(4,1) = X_D

COORDT(4,2) = Y_D

COORDT(4,3) = Z_D +(FRAME-1)*zMAX

The order of D is determined as 1, an interframe transfer is recognized (Frame 1 to Frame 2) and recorded as

FRAME1(2) = FRAME+1 = 2

COORD1(2,3) = 0

COORD1(2,1) = X_D

COORD1(2,2) = Y_D

COORD1(2,4) = α_{22}

COORD1(2,5) = $\sin\beta_{22}$

The interframe counter is listed with

EVENT1(2) = 1

and vertex number of D recorded with

STARNO(2) = STAR = 4

and the number of interframe transfers is updated

FRMTRN = ARMTRN+1 = 2

The program then returns to an evaluation of the event scanning status.

The evaluation of the event scanning status, since all tracks leaving all vertices within FRAME 1 have been traced, results in the tape storage of all retrieved data with an identifying tape record number. After storing these results on tape, the arrays and variables

ORDER

COORDT

COORDA

are initialized with zero values.

The program returns to the edge scanning mode and searches for further track entries. For the hypothetical event under study, the exit point in frame 1, X_2, is detected as an entry point -- although it is actually a point of exit with respect to the convention established during the tracing process. Accordingly, this entry point is treated as the start of a new event. The results for this "new" event, i.e., STAR, COORDT, ..., etc. are

EVENT = 2	the second event encountered
STAR = 1	designates the vertex X_2
COORDT(1,1)	x component of X_2
COORDT(1,2)	y component of X_2
COORDT(1,3)	z component of X_2

The vertex X_2 undergoes analyses which results in the detection of a track leaving X_2. This track is then followed until it reaches the vertex at G, the interframe transfer from Frame 1 to Frame 2. The evaluation of the scanning status for this "new" event recognizes that all tracks leaving all vertices of Event 2 have been exhausted, the pertinent data for Event 2 is stored on tape with an identifying record number and the program returns to search for new track entries.

Since there are no further entries to be detected the program then selects Frame 2 as the next frame to be investigated, Figures (III-18) and (III-19).

From Figure (III-18), we see that

FRAME = LSTFRM+1 = 2

LSTFRM = 2

EDGFLG = 1

Since the new frame number, 2, is less than the total number of frames, Frame 2 will be mounted. With Frame 2 mounted and readied, the program first investigates surface track entries, by examination of the interframe list, and then searches for track entries of new events originating on the edges of Frame 2.

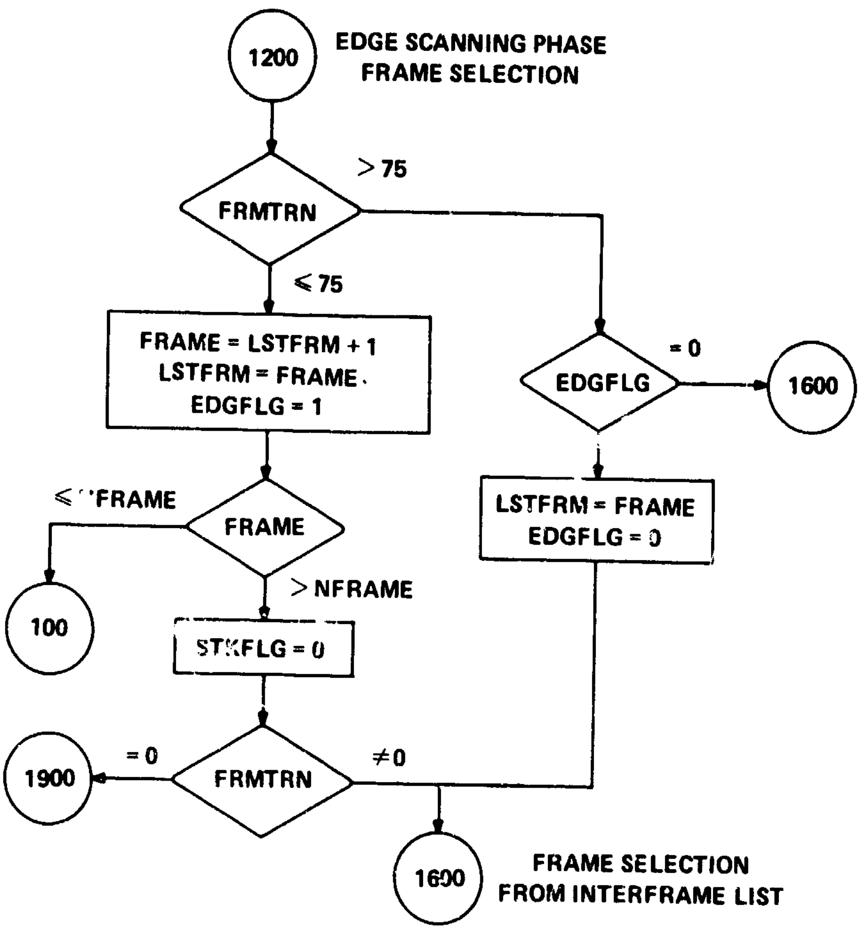


FIGURE (III-18) - EDGE SCANNING PHASE OF FRAME SELECTION

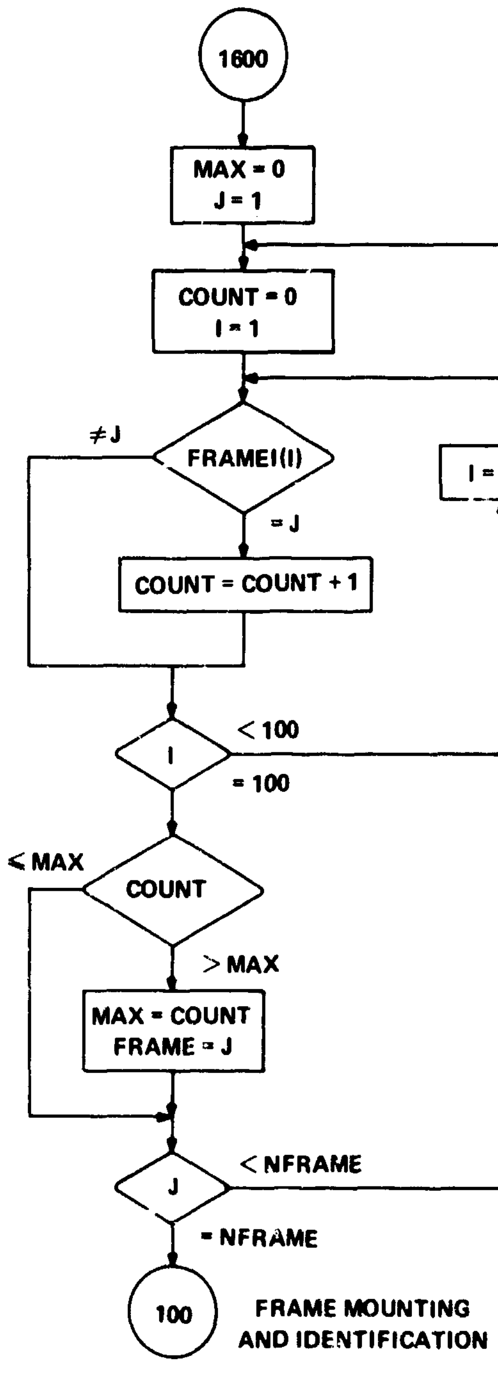


FIGURE (III-19) - FRAME SELECTION FROM INTERFRAME LIST

Before continuing with the trace of the program for our hypothetical event, pertinent information from Frame 1 is summarized in the table below.

	VERTEX NO.	VERTEX	ORDER	CHARACTERISTIC
Event 1				
	1	A	1	Entry point
	2	B	3	Internal Vertex
	3	C	1	Interframe Transfer
	4	D	1	Interframe Transfer
Event 2				
	1	X_2	1	Entry Point
	2	G	1	Interframe Transfer

Interframe List: Track Entries into Frame 2 from Frame 1

EVENT 1	EVENT 1	EVENT 2
FRAME1(1) = 2	FRAME1(2) = 2	FRAME1(1) = 1
COORD1(1,1) = X_C	COORD1(2,1) = X_D	COORD1(1,1) = X_G
COORD1(1,2) = Y_C	COORD1(2,2) = Y_D	COORD1(1,2) = Y_G
COORD1(1,3) = 0	COORD1(2,3) = 0	COORD1(1,3) = 0
COORD1(1,4) = α_{21}	COORD1(2,4) = α_{22}	COORD1(1,4) = α_G
COORD1(1,5)=$\sin\beta_{21}$	COORD1(2,5)=$\sin\beta_{22}$	COORD1(1,5)=$\sin\beta_G$
EVENT1(1) = 1	EVENT1(2) = 1	EVENT1(3) = 2
STARNO(1) = 3	STARNO(2) = 4	STARNO(1) = 1
	FRMTRN = 2	FRMTRN = 1

In Figure (III-9), Frame 2 is identified and those variables pertinent to the edge scanning mode are initialized. The interframe transfer list is examined with the program recognizing the existence of surface track entries.

The frame surface track entries are identified, Figure (III-20), and recorded as,

I1 = 1

EVENT = EVENT1(1) = 1

X_0 = COORD1(I1,1) = X_C

Y_0 = COORD1(I1,2) = Y_C

Z_0 = COORD1(I1,3) = Z_C

ALPHAT = COORD1(I1,4) = α_{21}

SINBTT = COORD1(I1,4) = $\sin\beta_{21}$

The interframe reference is then deleted, so that only those references that remain are to be considered during any further investigations of surface track entries, accordingly;

FRAME1(I1) = 0

FRMTRN = 1

The analysis of the vertex at C, Frame 2, results in the detection of a track leaving C. The result of the analysis is summarized by

ALPHA(1) = α_{21} SINBT(1) = $\sin\beta_{21}$

ALPHA(1) = 0, 1>1 SINBT(1) = 0, 1>1

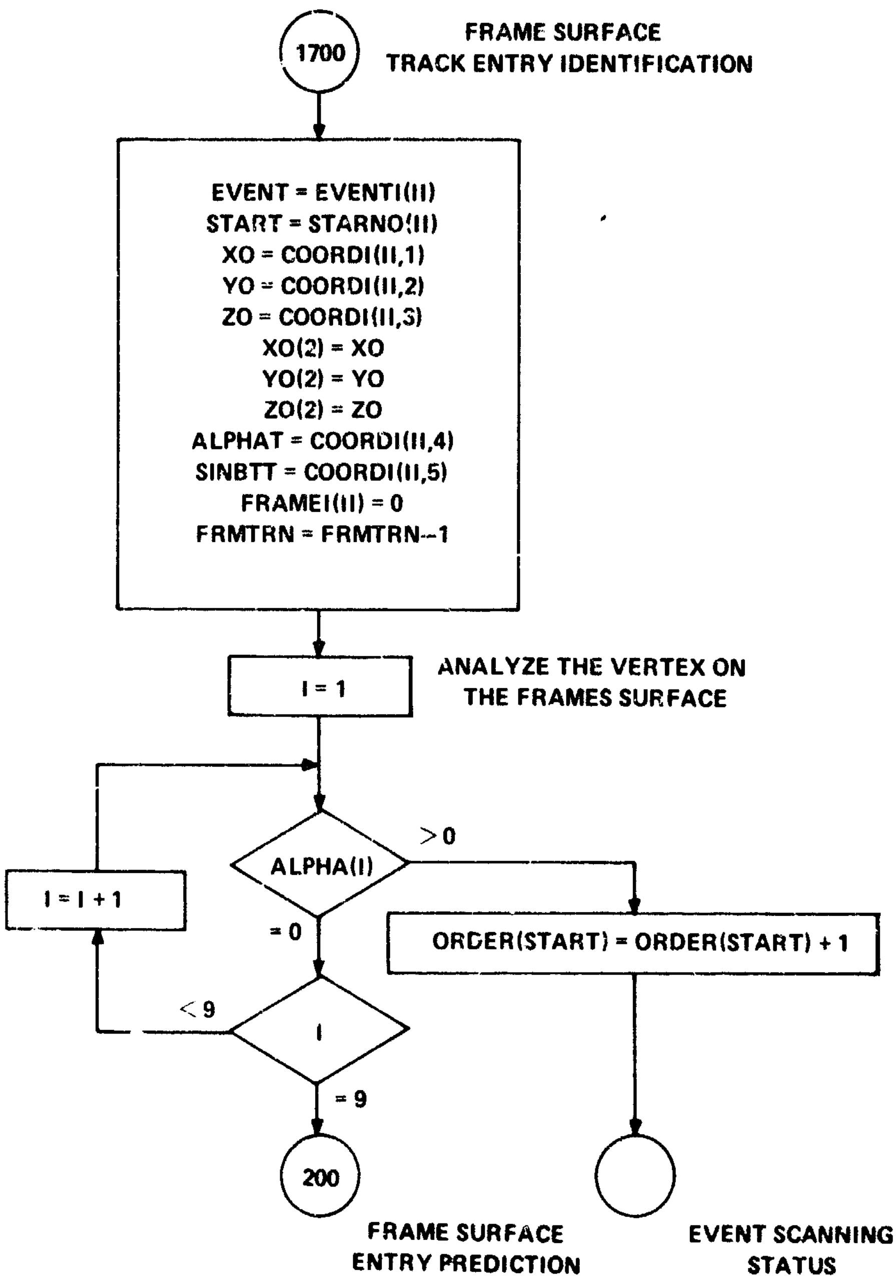


FIGURE (III 20) FRAME SURFACE TRACK ENTRIES

The non-zero value, ALPHA(1), indicates that the order of vertex C is computed as

ORDER(3) = ORDER(START) = 2

Continuing in Figure (III-13), the vertex number of C is designated with a STAR value

STAR = 1

and its coordinates stored in the COORDT array

COORDT(STAR,1) = $X_0 = X_C$

COORDT(STAR,2) = $Y_0 = Y_C$

COORDT(STAR,3) = Z_0 + (FRAME-1)*ZMAX

The tracks leaving C are tested for complementation, resulting with

ORDER(1) = 1

COORDA(1,1,1) = ALPHA(1) = α_{21}

COORDA(1,2,1) = SINBT(1) = $\sin\beta_{21}$

C is tested as a possible exit point, which it isn't; the interframe test is passed through and the scanning status is evaluated. The evaluation yields

START = 1

COUNT = 1

ALPHA+ = COORDA(1,1,COUNT) = α_{21}

SINBTT = COORDA(1,2,COUNT) = $\sin\beta_{21}$

for the track leaving C, in the surface of Frame 2, and this track is traced by the track following routine. The vertex at M is reached; it is analyzed and three tracks leaving M are detected.

ALPHA(1) = α_{M1} SINBT(1) = $\sin\beta_{M1}$

ALPHA(2) = α_{M2} SINBT(2) = $\sin\beta_{M2}$

ALPHA(3) = α_{M3} SINBT(3) = $\sin\beta_{M2}$

The new vertex is listed, designating the vertex with a STAR value, and recording its coordinates.

STAR = 2

COORDT(STAR,1) = X_M

COORDT(STAR,2) = Y_M

COORDT(STAR,3) = Z_M

The third track, defined by α_{M3} and $\sin\beta_{M3}$, is eliminated from consideration since it is the complement for the original track entering M. Accordingly,

COORDA(STAR,1,1) = α_{M1}

COORDA(STAR,2,1) = $\sin\beta_{M1}$

COORDA(STAR,1,2) = α_{M2}

COORDA(STAR,2,2) = $\sin\beta_{M2}$

ORDER(2) = 3

These tracks are evaluated by the program for their scanning status. The evaluated status dictates that the two tracks leaving M are followed

until the vertices, F and G, are reached. They are each identified, respectively with

STAR = 3, F

COORDT(STAR,1) = Y_F

COORDT(STAR,2) = Y_F

COORDT(STAR,3) = Z_F

and

STAR = 4, G

COORDT(STAR,1) = X_G

COORDT(STAR,2) = Y_G

COORDT(STAR,3) = Z_G

Vertex F is recognized as a point of penetration for the track descending Frame 2 into Frame 3, and is subsequently recorded in the interframe transfer lists,

FRAME1(1) = FRAME+1 = 3

COORD1(1,3) = 0

COORD1(1,1) = X_F

COORD1(1,2) = Y_F

COORD1(1,4) = α_{F1}

COORD1(1,5) = $\sin\beta_{F1}$

EVENT1(1) = 1

STARNO(1) = 3

FRMTRN = 2

Similarly, vertex G is recognized as a penetration point for the track ascending Frame 2 and into Frame 1, and is recorded in the interframe lists.

FRAME1(2) = FRAME-1 = 1

COORD1(2,3) = ZMAX

COORD1(2,1) = X_G

COORD1(2,2) = Y_G

COORD1(2,4) = α_{G1}

COORD1(2,5) = $\sin\beta_{G1}$

EVENT1(2) = 1

STARNO(1) = 4

FRMTRN = 3

After exhausting the tracing of event segments from Event 1, that event which entered Frame 2 through its surface, the program then proceeds to search for track entries along its edges, no further entries appear.

Following the edge search procedure for Frame 2, the third frame is then mounted for investigation. The track entries into the surface of Frame 3, vertices F and G, are treated, resulting in the detection of two tracks -- a track leaving F and exiting the edge at X_3 and a track leaving G and exiting at the edge at X_1.

The program then searches along the edge of Frame 3 where no new track entries are detected, however the exit points X_2 and X_3 were detected but only to be eliminated when compared against the list of exit points.

The program then examines all the data and recognizes that Events 1 and 2 are both one of the same, that is Event 2 designated the track which left the entry point X_2 and penetrated into the second frame. The program recognizes that this entry point X_2 is actually an exit point for Event 1.

III.10.0 CONCLUSIONS

The study demonstrated the viability of the conceptualization for a vidicon-optics hybrid computer system for a nuclear emulsive track tracer. Major emphasis, during the study, was placed on investigating scanning concepts, data assemblage, and their coordinated interaction in a realization of an automated system.

In order to accomplish the study's objectives -- demonstrating feasibility of vidicon-optics scanning concepts and data assemblage for particle track events -- an optically scaled breadboard model was constructed. This model, though crude, provided the means with which to evaluate, experimentally, the realizations for the concepts developed during the course of this study.

These experiments verified the success of the conceptual scanning regimes -- edge scanning, vertex analysis, and track following -- but of most importance they provided insight relevant to the hardware requirements for achieving any useful implementation of an automated track scanning system.

It is necessary to point out that the degree of resolution for any automated track scanner rests almost entirely with its hardware. Software can only treat the data produced by hardware, mathematical inference can, of course,

refine this data; but the ultimate bound or resolution is clearly limited by the hardware's performance. Our comments below relate to criteria for hardware selection.

The three major hardware aspects are the optical sensor, the optical sub-system, and the mechanical sub-system.

Present day optical sensors, e.g. vidicons, flying spot scanners, etc., have the capabilities to offer solutions of the order of 500 lines per inch and greater. Each of these sensors should be considered for speed of scanning and resolution.

The optical subsystem -- the collection of lenses which project real images of events onto the face of an optical sensor -- must be capable of providing variable magnifications with minimal distortion and variable depths of focus. These capabilities would provide faster track following, and greater resolutions in determing depth departure angles.

The mechanical subsystem relates to the mechanisms for controlling the motion of a two degree of freedom stage and motion along the z-axis -- the relative motion of the optical subsystem and sensor to the nuclear emulsion under investigation. The requirements for this system are quite severe; it must be capable of resolutions, in its movements, to the order of microns coupled with fast response times. These required resolutions imply that great care must be taken to mechanically shield the system against external distrubrances.

It is recommended that each of these three subsystems be evaluated in light of present day technology -- reviewing existing devices and systems -- including considerations for their interdependence with a system.

The software developed during this study is applicable with minor modifications to almost any realization of an automatic track scanning system. The "vertex-to-vertex" philosophy remains not only valid but appears to be a most efficient scheme for correlating the vast amounts of event data. Most modifications to the present software are envisioned in the form of hardware interface routines and editing programs which inject the principles of physics into evaluations of the data.